

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроніки
(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації
(повна назва кафедри)

«На правах рукопису»
УДК 621.396.2

«До захисту допущено»

Завідувач кафедри

(підпис) **Г.Г. Власюк**
(ініціали, прізвище)

“ ____ ” _____ 2018 р.

Магістерська дисертація

зі спеціальності 171 «Електроніка»

(код і назва спеціальності)

на тему: Особливості застосування спеціальних ефектів у процесі виробництва
мультимедійного контенту

Виконав: студент VI курсу, групи ДВ-71мп
(шифр групи)

Рябуха Олександр Тимурович
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник асистент, к.т.н. Філіпова Н.Ю.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає
запозичень з праць інших авторів без відповідних
посилань.

Студент _____

(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Інститут/факультет

Факультет електроніки
(повна назва)

Кафедра

звукотехніки та реєстрації інформації
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 171 «Електроніка» за спеціалізацією «Електронні та інформаційні системи і технології телебачення, кінематографії та звукотехніки»
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

**ЗАВДАННЯ
на магістерську дисертацію студенту
Рябуха Олександр Тимурович
(прізвище, ім'я, по батькові)**

1. Тема дисертації «Особливості застосування спеціальних ефектів у процесі виробництва мультимедійного контенту»

науковий керівник дисертації

Філіпова Наталія Юріївна к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2018 р. № 4114-с

2. Строк подання студентом дисертації 10 грудня 2018 р.

3. Об'єкт дослідження спеціальні ефекти засновані на методі системи частинок

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою) створення спеціального ефекту та відокремлення оптимального програмного середовища

5. Перелік завдань, які потрібно розробити: дослідити теоретичні аспекти розвитку та використання спеціальних ефектів, проаналізувати наявні програмні рішення, алгоритми та проблеми, практично дослідити ефективність існуючих рішень, дати

рекомендації щодо створення ефекту з бажаним результатом, розробити план-стартап проекту для поширення оптимального програмного середовища

6. Перелік графічного (ілюстративного) матеріалу 1) Тема, актуальність, мета та задачі; 2) Класифікації та предмет дослідження; 3) Використання методу системи частинок; 4) Дослідження та проблеми; 5) Практична оцінка існуючих рішень; 6) Висновки по роботі. Публікації.

7. Орієнтовний перелік публікацій «Моделювання спецефектів польоту на основі системи частинок» / О.Т. Рябуха // IV Міжнародна науково-практична інтернет-конференція «НАУКА ТА ОСВІТА В УМОВАХ ТРАНСФОРМАЦІЇ СУСПІЛЬСТВА»

8. Дата видачі завдання 15.10.2018 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Написання першого розділу: «Технології спеціальних ефектів»	15.10.2017-15.03.2018	
2	Написання другого розділу: «Алгоритм створення моделі класу нечітких об'єктів»	15.05.2018-15.10.2018	
3	Написання третього розділу: «Створення спеціального візуального ефекту»	20.10.2018-05.11.2018	
4	Написання четвертого розділу: «Розроблення стартап-проекту»	10.11.2018-25.11.2018	
5	Підготовка матеріалів до друку та оформлення пояснювальної записки	26.11.2018-01.12.2018	
6	Підготовка та оформлення презентації для доповіді	05.12.2018	

Студент

(підпис)

Рябуха О.Т.

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Філіпова Н.Ю.

(ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 113 с., 76 рис., 20 табл., 16 джерел, 1 додаток.

СПЕЦІАЛЬНІ ЕФЕКТИ, ВІЗУАЛЬНІ ЕФЕКТИ, КОМП'ЮТЕРНА ГРАФІКА,
СИСТЕМА ЧАСТИНОК, CGI, SPFX, SFX, VFX

При виробництві мультимедійного контенту все більша роль відводиться спеціальним ефектам. Розвиток комп'ютерних технологій вивело спецефекти на новий рівень. Тепер вони є потужним комерційним інструментом за допомогою якого не тільки втілюють у реальність фантастичні картини, а й керують увагою глядача, захоплюючи й зацікавлюючи.

Існує безліч програмних рішень для використання спеціальних ефектів.

Актуальним в цій сфері є найефективніше та економічно обґрунтовано сформулювати рішення поставленій задачі.

Об'єктом дослідження є спеціальні ефекти при виробництві мультимедійного контенту.

Метою дисертації є дослідження особливостей застосування спеціальних ефектів у процесі виробництва мультимедійного контенту, для досягнення поставленої мети необхідно вирішити наступні **завдання**: дослідити теоретичні аспекти розвитку та використання спеціальних ефектів, проаналізувати наявні програмні рішення, алгоритми та проблеми, практично дослідити ефективність існуючих рішень, дати рекомендації щодо створення ефекту з бажаним результатом, розробити план-стартап проекту для поширення оптимального програмного середовища.

Результатом дослідження є отримання деталізованої інформації щодо сучасних програмних середовищ, алгоритмів та методів створення спеціальних ефектів.

Предметом дослідження є програмне середовище для створення візуальних ефектів.

Методи дослідження: емпіричний метод, системний підхід та порівняльний аналіз

Наукова новизна одержаних результатів: Проаналізовано технології спеціальних ефектів в кіновиробництві а також наявні програмні рішення для створення комп'ютерної графіки такі як 3Ds Max, After Effect, Maya, Cinema 4D, Modo, Houdini, Softimage, LightWave, Blender.

Досліджено алгоритми створення моделі класу нечітких об'єктів в частині генерації, динаміки, візуалізації, ієрархії частинок.

Змодельовані спеціальні візуальний ефекту у вигляді полум'я в програмних середовищах: Magic Particles, Adobe After Effects, Blender.. Для створення ефекту при невеликих потужностях з високою реалістичністю доцільно використовувати програму Adobe After Effects, з на основі таких вмонтованих плагінів як Фрактальний шум і Система частинок з комбінацією таких інструментів як шари, маски та текстури.

Практичне значення одержаних результатів: Сформульовано методику створення полум'я в програмних середовищах: Magic Particles, Adobe After Effects, Blender та створено відеоряд з використанням системи частинок.

Надано порівняльну характеристику застосування цих програмних продуктів.

Публікація тез доповіді «Моделювання спецефектів польоту на основі системи частинок» на IV Міжнародній науково-практичній інтернет-конференції «НАУКА ТА

ОСВІТА В УМОВАХ ТРАНСФОРМАЦІЇ СУСПІЛЬСТВА» / Рябуха О.Т. : НТУУ
«КПІ ім. Ігоря Сікорського», 2018. – С.68-71.

THE SUMMARY

Dissertation contains: 113 pp., 76 dr.; 20 tb., 16 sources; 1 application.

SPECIAL EFFECTS, VISUAL EFFECTS, COMPUTER GRAPHICS, PARTICLE SYSTEM, CGI, SPFX, SFX, VFX.

In the master's thesis the features of the application of special effects in the production of multimedia content are analyzed.

The technology of special effects in cinema production is analyzed, as well as existing software solutions for creating computer graphics such as 3Ds Max, After Effect, Maya, Cinema 4D, Modo, Houdini, Softimage, LightWave, Blender.

The algorithms of creation of model of a class of fuzzy objects in the part of generation, dynamics, visualization, particle hierarchy are explored.

Formulated the methodology for creating flames in software environments: Magic Particles, Adobe After Effects, Blender, and created a video game using a particle system.

A project plan for the distribution of an optimal software environment has been developed.

ЗМІСТ

ПЕРЛІК УМОВНИХ СКОРОЧЕНЬ.....	10
ВСТУП.....	11
1 ТЕХНОЛОГІЇ СПЕЦІАЛЬНИХ ЕФЕКТІВ.....	13
1.1 Розвиток спеціальних візуальних ефектів	13
1.2 Галузі використання спеціальних ефектів	17
1.3 Програмні інструменти	20
2 АЛГОРИТМ СТВОРЕННЯ МОДЕЛІ КЛАСУ НЕЧІТКИХ ОБ'ЄКТІВ	34
2.1 Методика моделювання класу нечітких об'єктів	34
2.2 Алгоритм системи частинок	36
2.3 Генерація часток	38
2.4 Атрибути частинок	40
2.5 Динаміка частинок	42
2.6 Візуалізація частинок	43
2.7 Ієрархія частинок	44
2.8 Методи	45
2.9 Оптимізація	49
3 СТВОРЕННЯ СПЕЦІАЛЬНОГО ВІЗУАЛЬНОГО ЕФЕКТУ	55
3.1 Створення ефекту полум'я у програмі <i>Magic Particles</i>	55
3.2 Створення ефекту полум'я у програмі <i>After Effects</i> на основі системи частинок - CC Particle World	61

3.2.1 Створення ефекту полум'я на основі системи частинок - CC Particle World з використанням типу частинки – Затінена сфера	64
3.2.2 Створення ефекту полум'я на основі системи частинок - CC Particle World з використанням типу частинки – Лінза-бульбашка	67
3.3 Створення ефекту полум'я у програмі <i>After Effects</i> на основі ефектів Фрактальний шум - Fractal Noise, з допомогою масок та шарів та з використанням системи частинок - CC Particle World	71
3.4 Створення ефекту полум'я у програмі <i>Blender</i>	80
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	90
4.1 Загальні відомості	90
4.2 Технологічний аудит ідеї проекту	91
4.3 Аналіз ринкових можливостей запуску стартап-проекту	92
4.4 Розроблення ринкової стратегії проекту	97
4.5 Розроблення маркетингової програми стартап-проекту	99
ВИСНОВКИ	103
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	105
ДОДАТОК А	107

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПК	– персональний комп'ютер
CGI	– computer-generated imagery
CPU	– central processing unit
GPU	– graphics processing uni
PAL	– Physics Abstraction Layer
SPFX, SFX	– special effect
SPH	– Smoothed Particle Hydrodynamics
SWOT	– strengths, weaknesses, opportunities, threats
VFX	– Visual effects
3D	– 3-dimensional

ВСТУП

Візуальні ефекти грають велику роль у сучасних розповідях (storytelling), вони застосовуються не тільки в постановочних кінофільмах, але також і в телевізійних передачах, комп'ютерних іграх і в парках атракціонів. Спеціальні візуальні ефекти стають все більш реалістичними на основі застосування сучасних потужних процесорів, методів комп'ютерної графіки і технологій відображення. Для створення комп'ютерної графіки використовують безліч різних додатків, в тому числі потужні універсальні 3D редактори. Вони відрізняються між собою специфікою роботи, можливостями та сферою застосування.

Однак існує проблема реалістичного відображення об'єктів нерегулярної форми, таких явищ, як хмари, дим, вода і вогонь, виявилось складним з існуючими методами синтезу комп'ютерного зображення. Останніми роками дослідники, згідно з різними вимогами, запропонували низку підходів до моделювання та відтворення. Ці підходи можна грубо класифікувати за двома основними способами: онтогенетична модель (об'ємне процедурне моделювання, метод фракталів, метод системи частинок) [10]. Моделювання системи часток безпосередньо імітує гранули і додає кожен гранулу з деякими параметрами, такими як розташування, розмір, колір тощо.

Останніми роками дослідники, згідно з різними вимогами, запропонували низку підходів до моделювання та відтворення хмар. Ці підходи можна грубо класифікувати за двома основними способами: онтогенетична модель та фізично ґрунтована модель [10]. Онтогенетична модель включає об'ємне процедурне моделювання [12], метод фракталів [13, 14] та метод системи частинок [10]. Принцип об'ємного процедурного моделювання базується на стільникових автоматах. Цей метод має об'ємний розрахунок, і важко провести моделювання в реальному часі, оскільки він використовує ту ж техніку рендерінга на кожному окремому символі в сцені. Фрактальний метод використовує математичне реляційне вираження для створення

геометричних форм, придатних для точної та статичної графіки. Порівняно з двома вищезазначеними способами, моделювання системи часток безпосередньо імітує гранули і додає кожну гранулу з деякими параметрами, такими як розташування, розмір, колір тощо. Система частинок показує динамічну випадкову особливість нечіткого об'єкта, і використовується для моделювання нечітких об'єктів, таких як сніг, дощ, хмара та реактивний вогонь. Перевага цього методу полягає в легкості контролю форми частинок та в тому, що він реалізований в реальному часі [7 – 9].

Найдавніші комп'ютерні відеоігри зображували вибухи космічних кораблів з безліччю маленьких сяючих точок, які покривають екран [8]. Точкові джерела використовувалися в якості типу графічних даних у багатьох системах тривимірного моделювання (наприклад, ранні льотні тренажери Еванса і Сазерленда), хоча в літературі є кілька реальних посилань. Роджер Уілсон в штаті Огайо [4] використовував частки для моделювання диму, що виходить з димової труби. У його моделі не було ні стохастичного контролю, ні динаміки. Alvy Ray Smith і Jim Blinn використовували частки для моделювання створення зірки і смерті в галактиках для серії Cosmos [11]. Алан Нортон [9] використовував частки для генерації і відображення тривимірних фрактальних форм. Джим Блинн [3] обговорював функції відбиття світла для імітації світла, що проходить крізь нього і відбивається шарами частинок. Його техніка використовувалася для створення зображень кілець Сатурна. Об'ємні подання також були запропоновані в якості життєздатних альтернатив поверхневим зображенням.

1 ТЕХНОЛОГІЇ СПЕЦІАЛЬНИХ ЕФЕКТІВ

В наш час бурхливо розвиваються технології використання спеціальних візуальних ефектів

Візуальні ефекти грають велику роль у сучасних розповідях (storytelling), вони застосовуються не тільки в постановочних кінофільмах, але також і в телевізійних передачах, комп'ютерних іграх і в парках атракціонів. Спеціальні візуальні ефекти стають все більш реалістичними на основі застосування сучасних потужних процесорів, методів комп'ютерної графіки і технологій відображення.

Однак вимоги до якості і інтерактивності дуже розрізняються залежно від ситуації. У кінофільмах потрібно виключно висока якість візуальних ефектів, для чого зазвичай використовуються надзвичайно потужні комп'ютерні засоби. Візуальні ефекти в телевізійних передачах і комп'ютерних іграх повинні забезпечуватися з частотою не менш 25-60 Гц. У випадку комп'ютерних ігор цього можливо досягти на звичайних домашніх ПК, ігрових приставках і все частіше в мобільних телефонах. У багатьох паркових атракціонах візуальні ефекти комбінуються зі спеціальними ефектами. Як і в комп'ютерних іграх, в деяких розважальних атракціонах за допомогою візуальних ефектів підтримується інтерактивність. Але, на відміну від ігор, телебачення та кінофільмів, вони можуть надати учасникам відчуття, які неможливо забезпечити при демонстрації візуальних ефектів на плоскому екрані.

1.1 Розвиток спеціальних візуальних ефектів

Якщо поглянути на історію кінематографії, то можна виділити три різних періоду розвитку візуальних ефектів:

1980-і pp. були «золотим століттям» реальних спеціальних ефектів. При зйомці сцен, за допомогою спеціальних масок, закривалися ділянки кадру, де буде малюнок.

Далі, за допомогою художників і подвійний експозиції, виходили комбіновані кадри (див. Рис.1.1). Це десятиліття почалося з фільмів «Імперія завдає удару у відповідь» («The Empire Strikes Back») і «Інопланетянин» («ET: The Extra-Terrestrial»), які показали, чого можливо домогтися при використанні складних моделей, макетів і систем управління рухом кінокамер, і закінчилося блокбастерами зі спецефектами «Бетмен» («Batman») і «Індіана Джонс і останній хрестовий похід» («Indiana Jones and the Last Crusade»).

Першими хто об'єднав камеру і комп'ютер була компанія Lucasfilm.

Сучасні комп'ютерні зображення зобов'язані першому підрозділу комп'ютерної графіки який в 1982 році підтвердив свою кваліфікацію роботою над спеціальними ефектами для фільму «Зоряний шлях 2» («Star Trek II: The Wrath of Khan»). Створюючи обладнання та програмне забезпечення вони витратили п'ять місяців на вдосконалення технік текстурування, розфокусовки, цифрового монтування і композиційних програм щоб створити для кіно перші комп'ютерні спецефекти.



Рисунок 1.1 – Комбінований кадр із фільму «Той, що біжить по лезом» (1981)

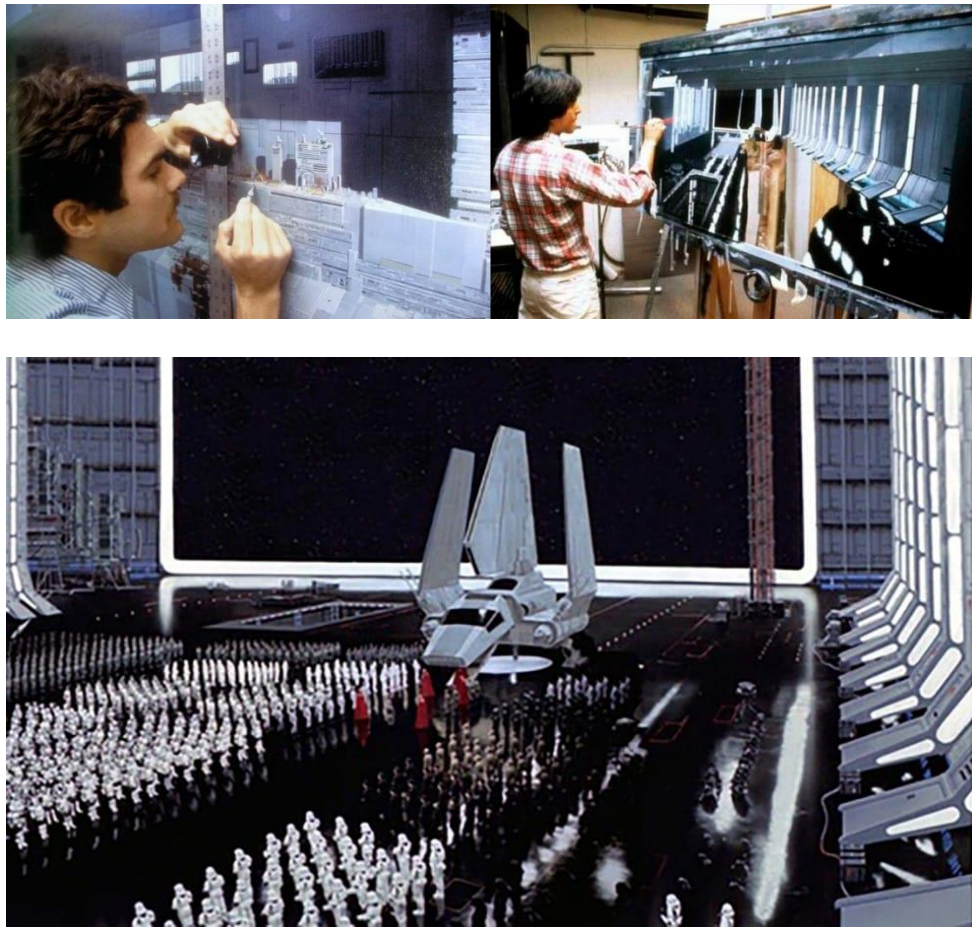


Рисунок 1.2 – Використання макету при роботі над фільмом «Зоряні війни»

У 1985 році з'явився перший комп'ютерний персонаж у фільмі «Молодий Шерлок Холмс» («Young Sherlock Holmes»).

1990-і *pp.* ознаменували розквітом ефектів CGI (computer-generated imagery, комп'ютерна анімація): постійно зростаюча потужність комп'ютерів привела до створення інструментальних засобів, які дозволили розробникам візуальних ефектів робити речі, неможливі при застосуванні фізичних методів. У фільмах, подібних до «Термінатор 2: Судний день» («Terminator 2: Judgment Day»), використовувався великі об'єми комп'ютерної анімації поряд з традиційними ефектами. Термінатор Арнольда Шварценеггера (Arnold Schwarzenegger) моделі 101 створювався за допомогою гриму і складного макетування, але вже більш досконалий зразок моделі

T-1000, який грав Роберт Патрік (Robert Patrick), був породжений комп'ютерною графікою.

З фільмом «Парк Юрського періоду» («Jurassic Park», 1993 г.) відбулася революція комп'ютерної графіки. А закріпилася вона в «Форрест Гамп» («Forrest Gump», 1994 г.). Була використана маніпуляція з мімікою, рухом, фоном та ін. для задання потрібного настрою. Спецефекти почали використовувати у всіх жанрах. Це і стало причиною такого швидкого розвитку і величезного спектру вибору інструментів і програмного забезпечення.

Метод системи частинок, що розглядається в цій роботі, почали використовувати під час роботи над фільмом «Смерч» («Twister», 1996 г.) - з'явилася необхідність відтворити уламки, пил, водні бризки комп'ютерною графікою та за допомогою спеціальних ефектів об'єднати плани.

В останні десятиліття використання CGI швидко розширювалося і поглиблювалося, а традиційні ефекти приходили в занепад. Наприклад трилогія «Володар перстнів» («Lord of the Rings») і фільм «Трансформери» («Transformers») являють собою наочні приклади домінування CGI. У «Трансформерах» спостерігаються трансформації, задіяні одночасно десятками тисяч рухомих частин, які взаємодіяли між собою і навколишнім світом (див. Рис.1.3).

У цифрову епоху в арсеналі художників з'явилося неймовірна кількість інструментарію для створення візуальних ефектів. Зараз, мабуть, ми не знайдемо кіно, в якому б не застосовувалися технології візуальних ефектів. Вони використовуються навіть там, де ми не підозрюємо їх присутність. Це стало невід'ємною частиною оповідання і потужним інструментом посилення історії в кіно [2].

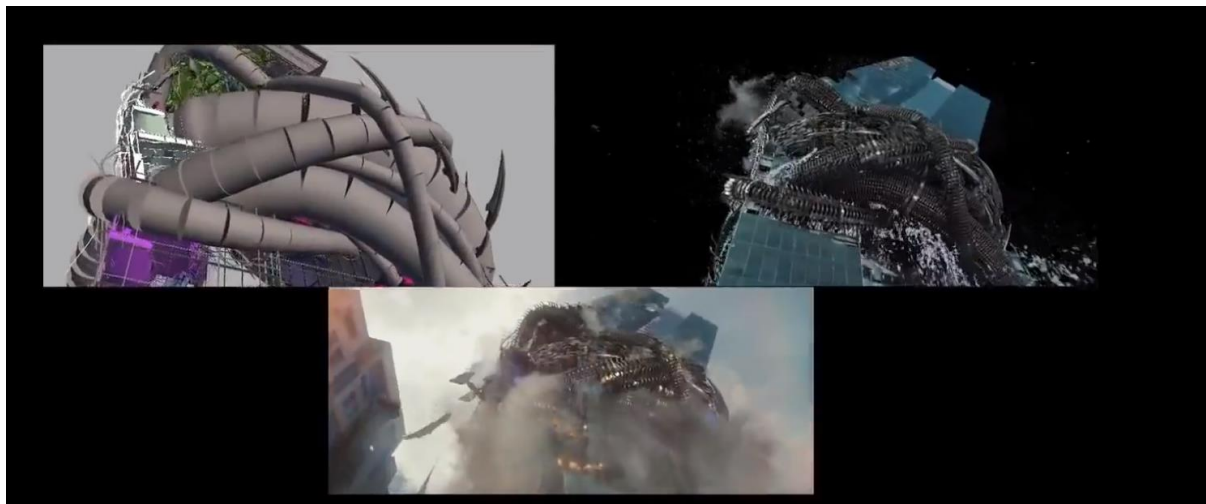


Рисунок 1.3 – Комп’ютерна графіка у фільмі «Трансформери»

Сьогодні художнє маскування (matte painting) - один з безлічі VFX (Visual effects) інструментів, який закриває тільки малу частину стадії розробки (pipeline). Індустрія включає в себе величезну кількість вузькоспеціалізованих професій - художники, аніматори, моделлери, скульптори, візуалізатори, композери і колористи. Є окремі департаменти, які займаються тільки шерстю і волоссям для CGI персонажів. Також не варто забувати, що інструментарій теж змінився. І якщо раніше художники малювали на склі, а потім перейшли в графічні редактори, то сьогодні, в основному, використовується тривимірна графіка, робота з частинками, симуляція, фотограмметрія та ще тонна технологій для досягнення реалістичною графіки. Адже саме максимальна реалістичність картини дає ту атмосферу, яка занурить глядача в історію, яка відбувається перед його очима.

1.2 Галузі використання спеціальних ефектів

Спецефект, спеціальний ефект (англ. special effect, скор. SPFX, SFX або FX) - технології кінематографа і телебачення, що підвищують вірогідність і виразність

екранного зображення. До спецефектів також відносять оптичні ілюзії і трюки, що дозволяють створювати віртуальний світ в індустрії відеоігор і симуляторів.

До візуальних спецефектів відносяться комбіновані зйомки, а також комп'ютерну графіку.

Спеціальні ефекти, тобто графіка, генерована з використанням комп'ютерів і відтворена в складі реальної сцени, тепер є загальнопоширеним компонентом телевізійних програм. Починаючи з найпростішої карти погоди, що вставляється на блакитний екран за спиною ведучого програми, і закінчуючи складною тривимірною графікою, яка б показала положення бігунів на дистанції або ситуацію «поза грою» на футбольних змаганнях. Ця графіка допомагає виробникам телевізійних програм донести для телеглядачів інформацію в простій для розуміння і візуально наочній формі. Крім того, віртуальна графіка використовується як творчий інструмент, на якому можуть ґрунтуватися нові види програм. Використання цієї графіки може дозволити скоротити вартість виробництва звичайних програм за рахунок заміни частин декорацій комп'ютерними зображеннями.

Нові розробки обіцяють ще більше розширити області проникнення тривимірної графіки і допомогти розвитку крос-платформного контенту, оскільки 3D-графіка завжди була і залишається основною опорою індустрії розваг. Хоча більшість елементів технології, що використовується для додавання віртуальної графіки до телевізійних програм, є тими ж, що використовуються в індустрії кінематографії, їх використання на телебаченні суттєво відрізняється швидкістю виробництва і рівнем досконалості. У той час як на створення кінофільму зазвичай витрачається багато часу, і бюджет дозволяє підтримувати багатомісячне комп'ютерне редагування, при виробництві телевізійних програм бюджет рідко дозволяє підтримувати цей рівень редагування. Крім того, в даному випадку не потрібно проводити контент з високим рівнем розширення. Багато видів програм, які мають потребу у віртуальній графіці,

трансляються в реальному часі, або, принаймні, час від їх запису до трансляції не повинно перевищувати кілька годин.

Комп'ютерні ігри дали життя індустрії, яка постійно нарощує обсяг інтерактивного контенту і забезпечує нові враження від процесу гри. Періодично з'являються і рекламуються нові ігри. Крім нововведень в основної сюжетної лінії і режимах гри, в рекламі нових ігор часто згадуються візуальні ефекти. Наприклад, у гри Far Cry 2 компанії Ubisoft є багато цікавих особливостей, таких як зміна погодних умов, освітлення і т.д. Однією з відмінних рис цієї гри є вражаючий механізм поширення вогню, який забезпечує можливість нових вражень від процесу гри: гравці можуть використовувати в бою пляшки із запальною сумішшю і вогнемети; вогонь поширюється відповідно до законів природи і напрямком вітру, що, в кінцевому рахунку, може призводити до великих пожеж.

У широкому сенсі, в комп'ютерних іграх всі графічні ефекти є візуальними. Ці ефекти є основними елементами ігор при моделюванні реального або штучно створеного світу. Однак автори спираються на більш вузьке визначення візуальних ефектів, маючи на увазі графічні ефекти, які використовуються для спеціальних цілей. Ці ефекти часто моделюють деякі явища в людському житті і можуть викликати у гравця особливе емоційний відчуття за рахунок, наприклад, використання об'ємного туману для створення духу таємничості.

Спецефекти в комп'ютерних іграх часто повторюють ті, які вже застосовувалися в кіно, але з іншими обмеженнями на якість, інтерактивність і час візуалізації. Точка огляду для кожного кадру фільму є визначеною заздалегідь, і аудиторія не може на неї впливати. Результуючі зображення готуються в режимі офлайн на спеціальних кластерах і налаштовуються в кіностудіях на стадії редагування.

Для ігор потрібні спеціальні ефекти тільки прийнятної якості з дозволом приблизно $1,280 \times 720$ для персональних комп'ютерів або телевізорів. Гравці можуть

змінювати свою точку огляду. Графічні карти, вбудовані в персональний комп'ютер або ігрову приставку, синтезують кожен кадр в реальному часу. У кіноіндустрії на підготовку кадру можна витратити від кількох годин до декількох днів, а в відеоіграх частота зміни кадрів тепер становить від 1/25 до 1/60 секунди. При витраті кілька десятків мілісекунд на підготовку кожного кадру необхідно візуалізувати реалістичні спеціальні ефекти, на які можуть впливати гравці, які спостерігають за ними з будь-якої точки огляду. У порівнянні з офлайновими методами візуалізації, використовуваними в кіноіндустрії для створення візуальних ефектів, методи реального часу, необхідні для генерації ігрових візуальних ефектів, зазвичай забезпечують лише апроксимацію, спрямовану на скорочення обсягу обчислень в реальному часу за підтримки якості графіки на візуально прийнятному рівні.

1.3 Програмні інструменти

Для створення комп'ютерної графіки використовують безліч різних додатків. Умовно їх розділять на наступні групи:

- Програми для цифрового скульптінга (Pixologic ZBrush, Autodesk Mudbox).
- Ігрові двигуни (Unreal Engine 4, Unity 5, CryEngine 3).
- Вузькоспеціалізовані програми, спеціалізовані під конкретні завдання (анімація рідин - RealFlow, створення текстур - Mari та ін.).
- Універсальні 3D редактори (Cinema 4D, 3Ds Max, Maya, Houdini і т.д.).

Розглянемо універсальні 3D редактори (Full 3 D Suites).

Універсальні 3D редактори, як правило, містять все необхідне для CG: інструменти моделювання, анімації і візуалізації.

Вибір інструмента залежить від багатьох чинників: особистих переваг CG - художник, поставлених цілей, фінансових можливостей і т.д.

При виборі програми слід звернути увагу на наступне:

- функціонал програми;
- зручність користування (інтуїтивний інтерфейс і т.д.);
- доступність, ціна.

Більшість фахівців у своїй роботі використовують відразу декілька програм: деякі речі простіше і швидше робити в сторонніх додатках (деталізація, постобробка, симуляція і ін.).

Найпопулярніші 3D пакети: 3Ds Max, Adobe After Effects, Autodesk Maya, Cinema 4D, Modo, Side Effects Houdini, Softimage, LightWave.

3Ds Max

3Ds Max - серед 3D редакторів, дуже популярний інструмент, №1 у виборі багатьох початківців і просунутих фахівців. Займає провідні позиції в сфері дизайну та архітектурної візуалізації. Часто використовується в кіновиробництві та ігровій індустрії.

У цьому програмному середовищі варто виділити такі можливості:

- моделювання на основі полігонів, сплайнів і NURBS,
- потужна система частинок,
- модуль волосся / шерсть,
- розширені шейдери Shader FX,
- підтримка нових і вдосконалених механізмів Iray і mental ray,
- анімація натовпу,
- імпорт з Revit і SketchUp,
- інтеграція композітінга.

І багато інше.

Переваги: величезний функціонал, безліч плагінів і навчальної інформації.

Недоліки: не такий простий в освоєнні, так само цьому програмного забезпечення потрібні серйозні оновлення.

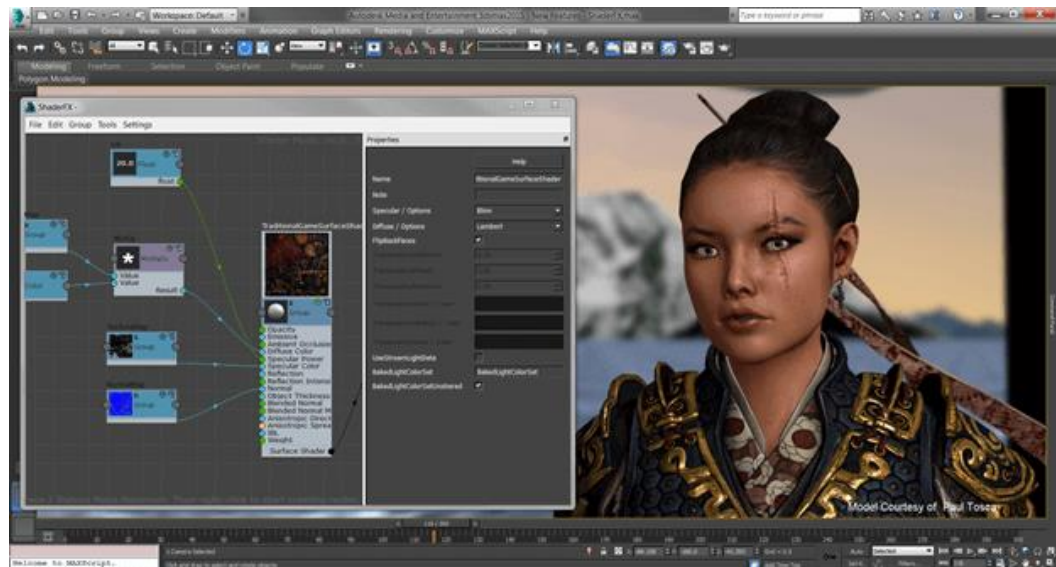


Рисунок 1.4 - Інтерфейс програми 3Ds Max

Adobe After Effects

Найпотужніший інструмент створення motion графіки. Так само використовується для композітінга.

Ось "Ходячих мерців", наприклад, роблять в *After Effects* (Про це розповів і показав Sam Nicholson з Stargate Studio на заході Adobe at NAB 2014 року). Можливості створення 3D графіки обмежені, водночас використовується для роботи з 3D об'єктами і простором.



Рисунок 1.5 - Створення сцени із серіалу «Ходячі мерці» в After Effects

Autodesk Maya

Maya - промисловий стандарт 3D графіки в кіно і телебаченні фахівців. Maya популярна серед великих студій масштабних проектів в рекламі, кіно, ігрової індустрії. Пакет орієнтований для створення анімації.

Варто виділити такі програмні можливості:

- повний набір інструментів для NURBS- і полігонального моделювання;
- потужні засоби загальної і персонажної анімації;
- розвинена система частинок;
- технологія Maya Fur (створення хутра, волосся, трави);
- технологія Maya Fluid Effects (моделювання рідин, атмосфери);
- динаміка твердих і м'яких тіл;
- широкий набір засобів створення динамічних спецефектів;
- UV-текстури, нормалізує обмін речовин і колірне кодування;

- багатопроцесорний гнучкий рендерінг.

Переваги: величезний функціонал і можливості.

Недоліки: тривале і складне навчання, високі вимоги до системи, висока ціна.



Рисунок 1.6 - Інтерфейс програми Autodesk Maya

Cinema 4D

Cinema 4D - один з найкращих і зручних 3D пакетів на сьогоднішній день. Величезний функціонал: від моделювання, анімації, ефектів до «ліплення» і модуля BodyPaint 3D. У Cinema 4D більш зрозумілий і зручний інтерфейс ніж у 3Ds Max і Maya. Це програмне середовище широко використовується в моушен-дизайні, кіноіндустрії та рекламі.

Можливості:

- полігональне і NURBS-моделювання;

- BodyPaint 3D (модуль для створення розгортки UV і карт текстур);
- генерація і анімація об'єктів;
- персонажної анімації;
- динаміка м'яких і твердих тіл;
- модуль для створення реалістичних волосся;
- система частинок Thinking Particles;
- непоганий вбудований візуалізатор.

Переваги: легкість в освоєнні, інтуїтивний інтерфейс, відмінний функціонал, безліч навчальних матеріалів, тісний зв'язок з Adobe After Effects, Houdini і т.д.

Недоліки: неузгоджена система переходу між версіями.

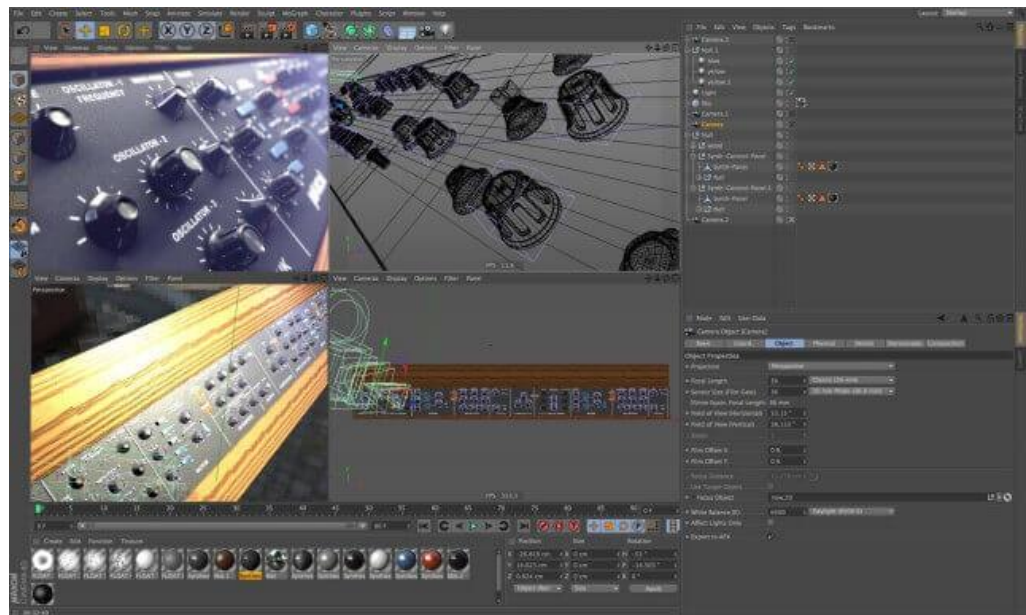


Рисунок 1.7 - Інтерфейс програми Cinema 4D

Modo

Modo - повноцінний продукт для моделювання, малювання, анімації і візуалізації. Включає також інструменти скульптинга і текстурного фарбування.

Завдяки зручності користування і високої продуктивності, у Modo репутація одного з найшвидших інструментів моделювання. Modo популярний в сфері реклами, розробки ігор, спецефектів і архітектурної візуалізації.

Можливості:

- полігональне і моделювання SDS;
- сучасні інструменти анімації;
- динаміка твердих і м'яких тіл;
- система малювання;
- матеріал Fur (хутро) для створення волосся, трави і хутра;
- інструменти ліплення;
- швидка і якісна візуалізація.



Рисунок 1.8 - Інтерфейс програми Modo

Переваги: потужний і зрозумілий інструментарій, висока продуктивність.

Недоліки: недостатньо інформації у відкритих джерелах.

Side Effects Houdini

Houdini - потужний професійний пакет для роботи з 3D графікою, в його основі процедурна, нодова система. Houdini ідеально підходить для створення складної динаміки, симуляції: частинок, рідини, диму, вогню, імітації природних явищ і т.д. А також це відмінний інструмент для створення вражаючих візуальних ефектів. Основна область застосування Houdini - кіноіндустрія.

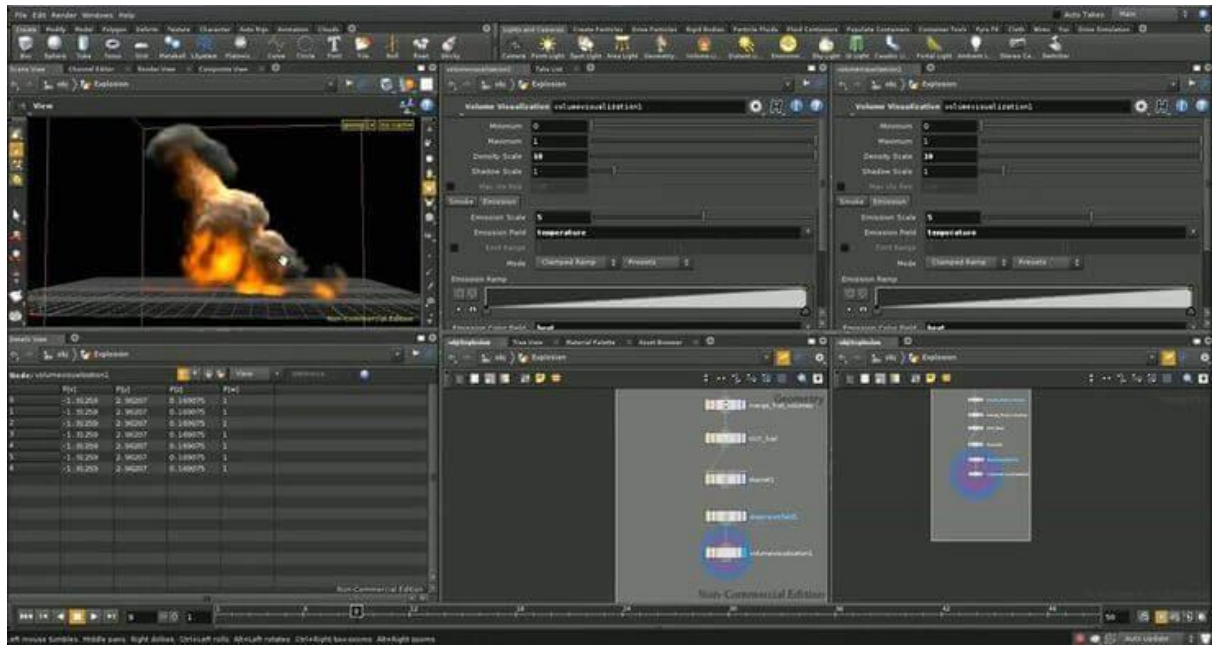


Рисунок 1.9 - Інтерфейс програми Houdini

Можливості:

- полігональне і NURBS-моделювання,
- анімація (ключова, процедурна),
- персонажна анімація,
- система частинок,
- динаміка твердих і м'яких тіл, тканин, вовни / волосся, газів і рідин,
- робота з об'ємним звуком,

- потужний рендер двигун Mantra,
- вбудований інструмент композітінга.

Переваги: висококласні спецефекти і анімація.

Недоліки: складна у використанні, висока ціна.

Softimage

Softimage (Autodesk Softimage, раніше Softimage / XSI) - програма для 3D анімації і створення візуальних ефектів в ігровій індустрії, кіно і телебаченні.

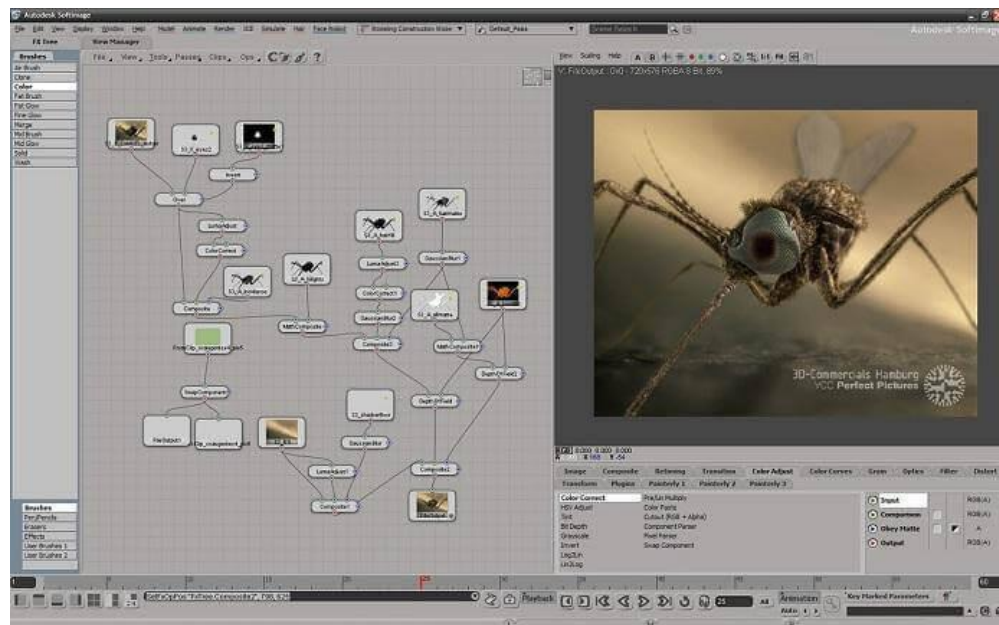


Рисунок 1.10 - Інтерфейс програми Softimage

У Softimage була одна з найкращих систем анімації. Завдяки унікальній системі ICE (Interactive Creative Environment - платформі візуального програмування, заснованої на нодах) пакет пропонує широку функціональність, гнучкість, високу продуктивність і якість.

Можливості:

- потужне полігональне, а також процедурне моделювання в середовищі ICE;
- фізика і динаміка частинок і геометрії;
- нелінійна анімація;
- інструменти лицьової анімації Autodesk Face Robot;
- вбудований MentalRay.

У 2008 році компанія Autodesk викупила Softimage у Avid за 35 млн. Доларів. У 2015 Autodesk оголосила про припинення продажів ліцензій на Softimage і фактично позбулася одного з найсильніших гравців на ринку. на офіційному сайті пропонується перейти на 3 Ds Max або Maya.

LightWave

Lightwave 3D - інструмент для 3D анімації та візуальних ефектів від компанії NewNek. З давніх-давен є промисловим стандартом в кіно і телебаченні.

Новий вдосконалений пакет LightWave 2015 пропонує величезні можливості: від динамічного моделювання, персонажної анімації, візуальних ефектів до розробки ігор і архітектурної візуалізації.

Можливості:

- інтуїтивний подвійний інтерфейс (modeler і layout);
- потужне полігональне моделювання;
- розвинена система анімації;
- система частинок;
- система спорядження персонажа Genoma 2;
- вдосконалений рендеринг;
- інтерактивне динамічне спадкування (Interactive Dynamic Parenting);

- гнучка система Bullet Dynamics;

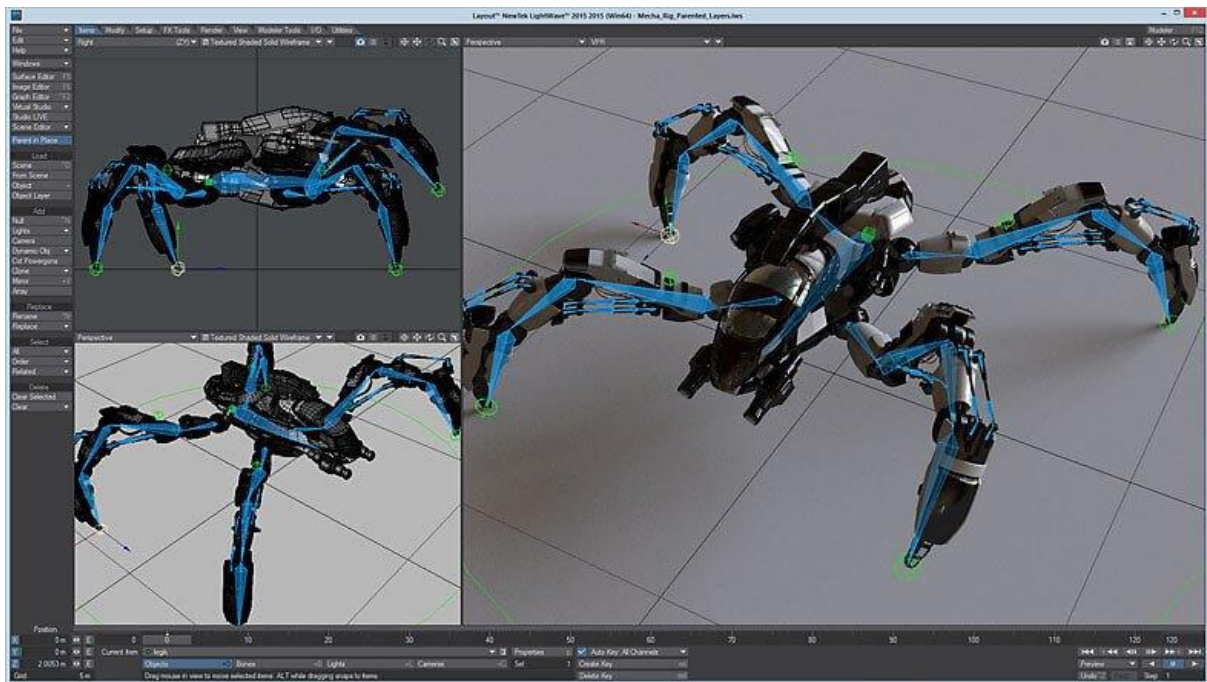


Рисунок 1.11 - Інтерфейс програми Lightwave 3D

Переваги: величезний функціонал, зручний подвійний інтерфейс.

Недоліки: не такий популярний в нашій країні, наявна проблема браку інформації.

Blender

Єдиний в списку безкоштовний 3D пакет, який практично не поступається по функціоналу платним програмам. Blender включає в себе інструменти для 3D моделювання, анімації, а також набір опцій для створення ігор, візуальних ефектів і скульптинга. Відмінна альтернатива «монстрам» 3D анімації. Завдяки підтримці Blender Foundation, програма дуже швидко і стабільно розвивається.

Можливості:

- полігональне моделювання, сплайни, NURBS-криві і поверхні;
- режим ліплення;
- система частинок;
- динаміка твердих і м'яких тіл: рідина, хутро/волосся і т.д.;
- скелетна анімація;
- вбудовані механізми рендерінга і інтеграція з сторонніми візуалізаторами;
- редактор відео;
- функції створення ігор і додатків (Game Blender).

Переваги: доступність, відкритий код, кросплатформеність, невеликий об'єм, широкий функціонал, можливість створення ігор.

Недоліки: відсутність документації в базовій поставці.

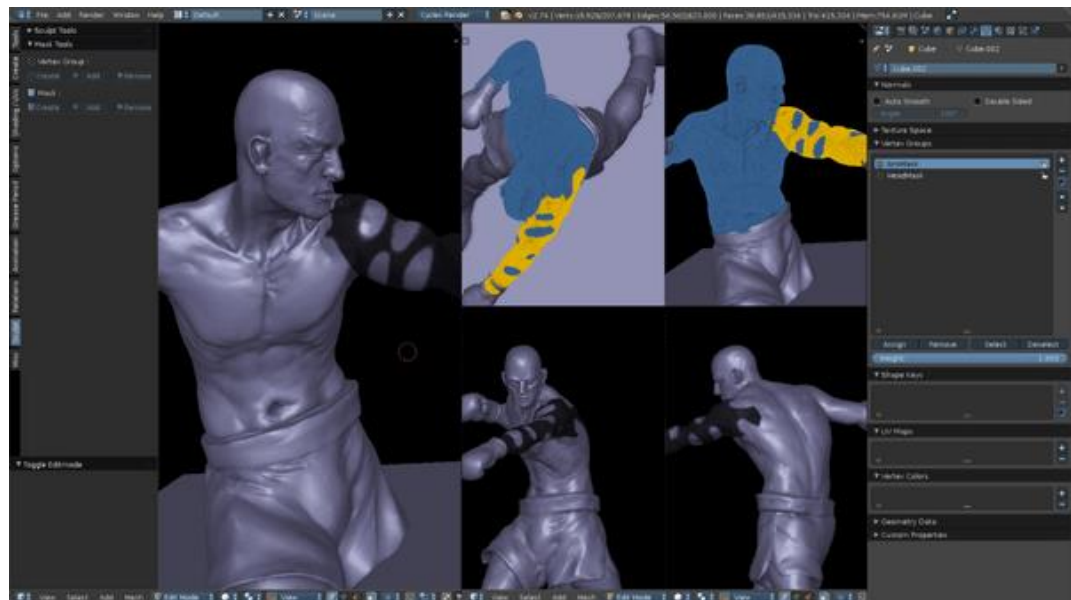


Рисунок 1.12 - Інтерфейс програми Blender

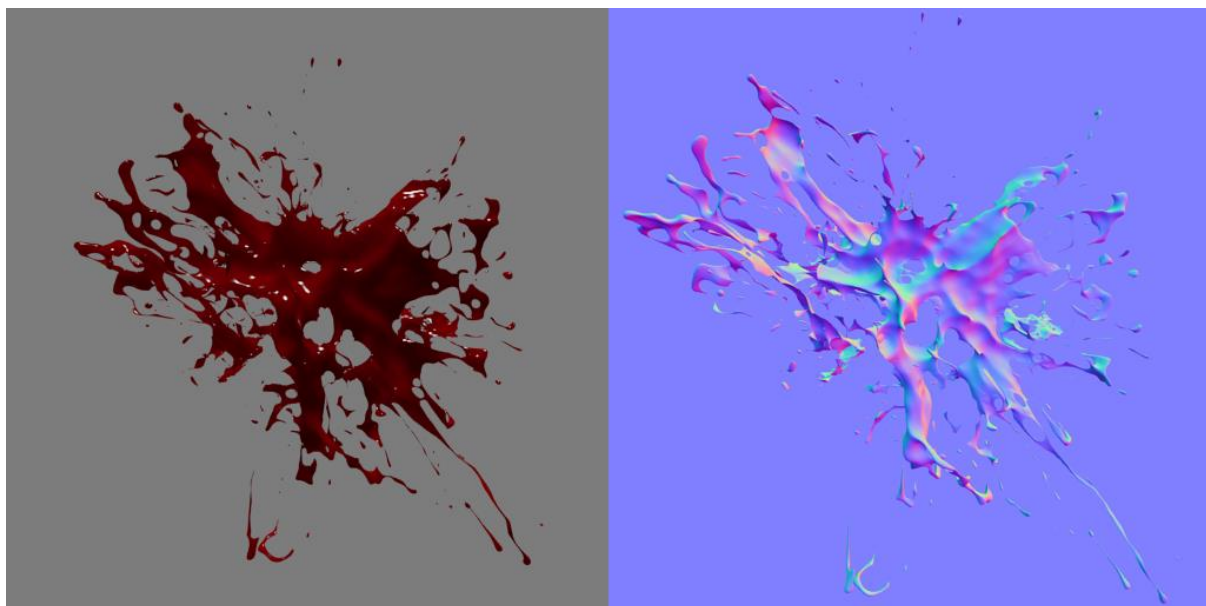


Рисунок 1.13 - Бризки крові, симульовані та відрендерені в Blender

Порівняльна таблиця найпопулярніших із них наведена нижче.

Таблиця 1.1 – Порівняння сфер застосування програмних інструментів

Програмне середовище	Комп'ютерні ігри	Спецефекти у кіновиробництві	Архітектурна візуалізація	Моушен-дизайн
3Ds Max	+	+	+	
After Effect		+		+
Maya	+	+		
Cinema 4D	+	+		+
Modo	+	+	+	+
Houdini		+	+	
Softimage	+	+		
LightWave	+	+	+	
Blender	+	+		

Висновки до розділу

1. Показано актуальність розробок спеціальних ефектів, галузі їх використання.
2. Показано, що потужні інструменти комп'ютерної графіки та спеціальних ефектів стають дедалі простіші у використанні та універсальні. Вони використовуються в кіноіндустрії, ігровій індустрії, як потужний візуалізатор для різного роду спеціалістів, на телебаченні, при створенні різноманітних відеокліпів: від музичних до рекламних і презентаційних.
3. Проаналізовано сфери застосування програмних інструментів таких як: 3Ds Max, After Effect, Maya, Cinema 4D, Modo, Houdini, Softimage, LightWave, Blender.

2 АЛГОРИТМ СТВОРЕННЯ МОДЕЛІ КЛАСУ НЕЧІТКИХ ОБ'ЄКТІВ

2.1 Методика моделювання класу нечітких об'єктів

Моделювання таких явищ, як хмари, дим, вода і вогонь, виявилось складним з існуючими методами синтезу комп'ютерного зображення. Ці «нечіткі» об'єкти не мають гладких, чітко визначених і блискучих поверхонь; замість цього їх поверхні є нерегулярними і складними. Вони не є жорсткими об'єктами, і їх руху не описуються простими перетвореннями, які поширені в комп'ютерній графіці.

Останніми роками дослідники, згідно з різними вимогами, запропонували низку підходів до моделювання та відтворення хмар. Ці підходи можна грубо класифікувати за двома основними способами: онтогенетична модель та фізично ґрунтована модель [10]. Онтогенетична модель включає об'ємне процедурне моделювання [12], метод фракталів [13] [14] та метод системи частинок [10]. Принцип об'ємного процедурного моделювання базується на стільникових автоматах. Цей метод має об'ємний розрахунок, і важко провести моделювання в реальному часі, оскільки він використовує ту ж техніку рендерінга на кожному окремому символі в сцені. Фрактальний метод використовує математичне реляційне вираження для створення геометричних форм, придатних для точної та статичної графіки. Порівняно з двома вищезазначеними способами, моделювання системи часток безпосередньо імітує гранули і додає кожену гранулу з деякими параметрами, такими як розташування, розмір, колір тощо. Система частинок показує динамічну випадкову особливість нечіткого об'єкта, і використовується для моделювання нечітких об'єктів, таких як сніг, дощ, хмара та реактивний вогонь. Перевага цього методу полягає в легкості контролю форми частинок та в тому, що він реалізований в реальному часі.

Подання систем частинок відрізняється трьома основними положеннями від уявлень, зазвичай використовуваних при синтезі зображень.

По-перше, об'єкт представлений не набором примітивних поверхневих елементів, таких як полігони або патчі, які визначають його кордон, а як хмари примітивних частинок, які визначають його обсяг.

По-друге, система частинок не є статичним об'єктом. Його частинки змінюють форму і рухаються з плином часу. Нові частки «народжуються», а старі частки «вмирають».

По-третє, об'єкт, як система частинок має нерегулярну форму, тобто форму яка не визначена повністю. Замість цього стохастичні процеси використовуються для створення і зміни форми і зовнішнього вигляду об'єкта.

При моделюванні нечітких об'єктів системний підхід частинок має кілька важливих переваг у порівнянні з класичними поверхнево-орієнтованими методами. По-перше, частка (на даний момент розглядається частинка як точка в тривимірному просторі) є набагато простішим примітивом, ніж багатокутник, найпростіший з поверхневих уявлень. Тому в тій же кількості часу обчислення можна обробляти більше основних примітивів і створювати більш складне зображення. Оскільки частка проста, вона також легко розмивається. Рушійне розмиття швидко рухомих об'єктів для усунення ефектів тимчасового згладжування в даний час в значній мірі ігнорується в синтезі комп'ютерних зображень. Друга перевага полягає в тому, що визначення моделі є процедурним і контролюється випадковими числами. Тому отримання високодетальної моделі не обов'язково вимагає великого часу людського проектування, як це часто буває з існуючими наземними системами. Система частинок може регулювати рівень деталізації відповідно до певних наборів параметрів перегляду. Як і в разі фрактальних поверхонь [5], збільшення масштабу в системі частинок може проявлятися все більш і більш детально. По-третє, об'єкти моделі систем частинок є «живими», тобто змінюють форму протягом певного періоду часу. Важко уявити складну динаміку цієї форми за допомогою методів поверхневого моделювання.

Найдавніші комп'ютерні відеоігри зображували вибухи космічних кораблів з безліччю маленьких сяючих точок, які покривають екран [8]. Точкові джерела використовувалися в якості типу графічних даних у багатьох системах тривимірного моделювання (наприклад, ранні льотні тренажери Еванса і Сазерленда), хоча в літературі є кілька реальних посилань. Роджер Уілсон в штаті Огайо [4] використовував частки для моделювання диму, що виходить з димової труби. У його моделі не було ні стохастичного контролю, ні динаміки. Alvy Ray Smith і Jim Blinn використовували частки для моделювання створення зірки і смерті в галактиках для серії Cosmos [11]. Алан Нортон [9] використовував частки для генерації і відображення тривимірних фрактальних форм. Джим Блинн [3] обговорював функції відбиття світла для імітації світла, що проходить крізь нього і відбивається шарами частинок. Його техніка використовувалася для створення зображень кілець Сатурна. Об'ємні подання також були запропоновані в якості життєздатних альтернатив поверхневим зображенням.

2. 2 Алгоритм системи частинок

Система частинок являє собою метод, який використовується для моделювання нерегулярного об'єкта графіки. Основа методології це визначення об'єкта системи частинок, який складається з великої кількості елементів, що мають неправильне переміщення і має функцію випадкового розподілу, кожна частка має час життя та інші властивості, такі як колір, форма, розмір, швидкість і тому подібні. Частинка змінює свою форму і стан руху, які могли б показати зміну загального вигляду сцени і динаміки.

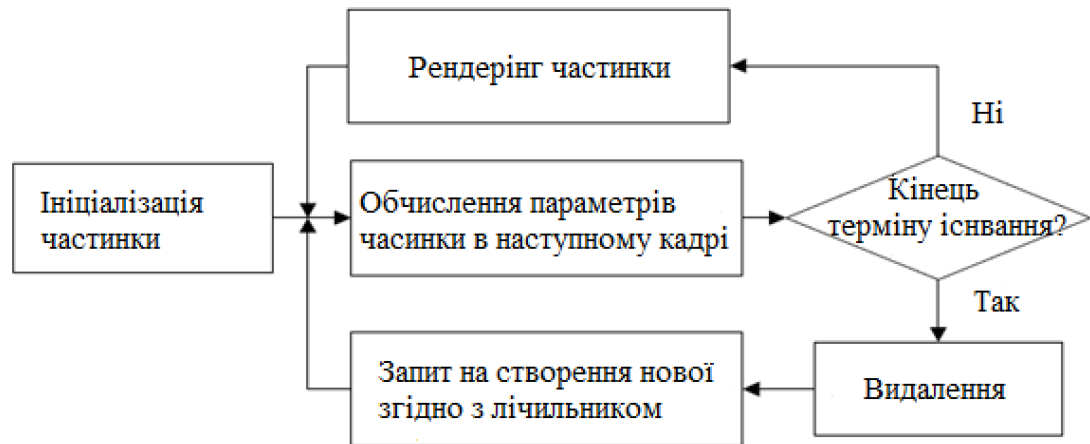


Рисунок 2.1 Схема моделювання об'єкта за допомогою системи частинок

Система частинок представлена динамічною випадковою функцією об'єкта без чітких меж, за допомогою якого можна моделювати такі нечіткі об'єкти як туман, дощ, дим і вогонь. На Рис.2.1 показано, як моделюється об'єкт за допомогою системи частинок.

Управління позицією і рухом системи частинок в просторі реалізовано за допомогою випромінювача (емітера), який виступає в ролі джерела частинок. У кожного емітера є набір властивостей, що визначають поведінку частинок в просторі. Математично кожна частка представляється як матеріальна точка з додатковими атрибутами, такими як швидкість, колір, орієнтація в просторі, кутова швидкість, і т.п. В ході роботи моделюються частки, кожна з яких змінює свій стан за певним, загальним для всіх частинок системи, закону, який встановлюється емітером [3]. Після проведення всіх розрахунків, частка візуалізується. Частка може бути представлена точкою, трикутником або повноцінної тривимірною моделлю.

Для обчислення кожного кадру в послідовності руху, виконується наступна послідовність кроків:

1. Нові частки створюються в системі.
2. Кожній новій частині присвоюється її окремі атрибути.
3. Частинки, час життя яких в рамках системи минув, видаляються з системи.
4. Інші частки переміщуються і перетворюються у відповідності з їх динамічними атрибутами.
5. Візуалізація.

Оскільки використовується процедурна модель, система може бути запрограмована для виконання будь-якого набору інструкцій в кожному кроці. Цей підхід може включати будь-які обчислювальні операції, що описують зовнішній вигляд або динаміку об'єкта. Наприклад, рух і перетворення частинок може бути прив'язане до вирішення системи диференціальних рівнянь.

2.3 Генерація часток

Частинки генеруються в систему частинок за допомогою контрольованих випадкових процесів. Один процес визначає кількість частинок, що надходять в систему протягом кожного інтервалу часу, тобто в даному кадрі. Число генерованих частинок важливо, оскільки воно сильно впливає на щільність нечіткого об'єкта.

Для вибору кількості нових частинок застосовують два методи.

У *першому методі* конструктор управляє середнім числом частинок, що генеруються в кадрі і його дисперсією. Фактичне число частинок, що генеруються в кадрі f , дорівнює

$$NParts_f = MeanParts_f + Rand() \times VarParts_f,$$

де $Rand()$ - процедура, яка повертає рівномірно розподілене випадкове число між -1.0 і +1.0, $MeanParts_f$ - середнє число частинок і $VarParts_f$ - його дисперсія.

У другому методі кількість нових частинок залежить від розміру екрана об'єкта.

Процедурна система частинок може визначати параметри представлення у певному кадрі, обчислювати приблизну площу екрану, яку він покриває, і відповідно встановлювати кількість нових частинок. Відповідне рівняння:

$$NParts_f = (MeanParts_{sa_f} + Rand() \times VarParts_{sa_f}) \times ScreenArea, \quad \text{де}$$

$MeanParts_{sa_f}$ - середнє значення для кожної області екрану, $VarParts_{sa_f}$ - його дисперсія і $ScreenArea$ - область екрану системи частинок.

Цей метод контролює рівень деталізації системи частинок і, отже, час, необхідний для відображення його зображення. Наприклад, немає необхідності генерувати 100 000 частинок в об'єкті, який покриває 4 пікселя на екрані.

Проста лінійна функція:

$$MeanParts_f = InitialMeanParts + DeltaMeanParts \times (f - f_0)$$

або ж

$$MeanParts_{sa_f} = InitialMeanParts_{sa} + DeltaMeanParts_{sa} \times (f - f_0), \quad \text{де } f$$

представляє поточний кадр, f_0 - перший кадр, протягом якого існує система частинок, $InitialMeanParts$ - середнє число частинок на першому кадрі, і $DeltaMeanParts$ - її швидкість зміни.

2.4 Атрибути частинок

Частинки генеруються через певний інтервал, який визначається на основі частоти генерації, початкові координати кожної частки визначаються положенням емітера і ступенем довільності. Емітер володіє наступними параметрами:

- положення в просторі
- час життя системи
- час життя частинки (задається діапазон мінімального і максимального значень)
- емісія

Кожна частка має ряд параметрів, що задаються властивостями системи:

- напрямок
- кут розкиду
- початкова швидкість
- гравітація
- радіальне прискорення
- тангенціальне прискорення
- розмір частки
- швидкість обертання
- альфа канал в началі і наприкінці життя частинки
- колір на початку і кінці життя частинки; задається в форматі RGB

Кілька параметрів системи частинок контролюють вихідне положення його частинок. Система частинок має положення в тривимірному просторі, яке визначає її початок. Два кута повороту навколо системи координат через це початок дають йому орієнтацію. Система частинок також має форму генерації, яка визначає область навколо її початку, в якій випадково ініціалізуються новостворені частки. Серед форм генерації реалізовано: сферу радіуса R , коло радіуса R в площині XY його системи

координат і прямокутник довжини L і ширину W в площині XY його системи координат. На Рис.2.3 показана типова система частинок зі сферичною формою генерації.

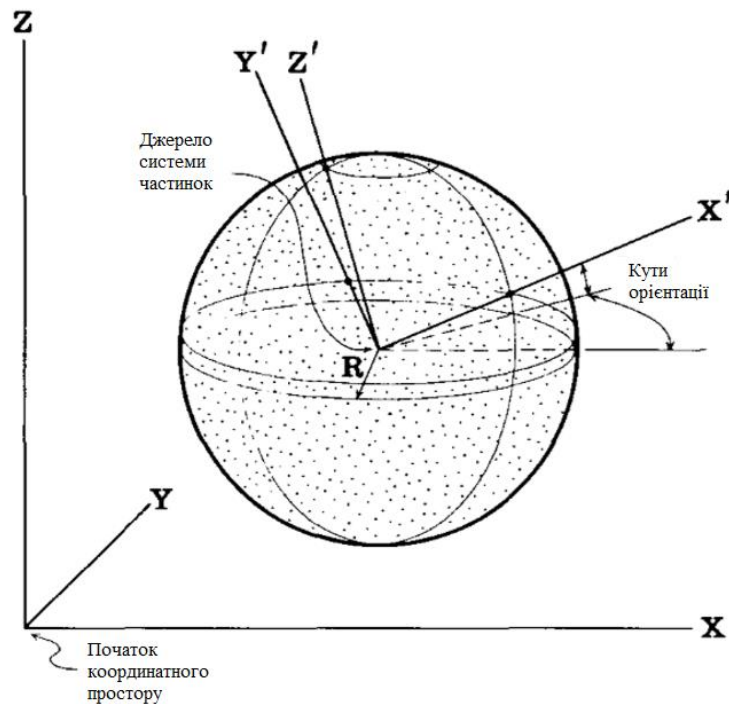


Рисунок 2.3 Типова система частинок з сферичною формою генерації

Форма генерації системи частинок також описує початковий напрямок руху нових частинок [6]. У формі сферичної генерації частинки рухаються назовні від початку системи частинок. В круговій або прямокутній формі частки переміщуються вгору від площини XY , але їм дозволено варіюватися від вертикалі відповідно до кута «викиду», що є іншим параметром (див. Рис.2.4).

Початкова швидкість частинки визначається

$$InitialSpeed = MeanSpeed + Rand() \times VarSpeed,$$

де *MeanSpeed* і *VarSpeed* - два інших параметра системи частинок, середня швидкість і дисперсія.

У кожному циклі етапу симуляції додаються нові частинки, видаляються старі і оновлюються параметри «живих» часток. Рух кожної частки засноване на математичній моделі руху матеріальних точок, де кожна частка виступає в якості цієї точки.

Кількість можливих параметрів управління атрибутами і їх варіантів нескінченно [7].

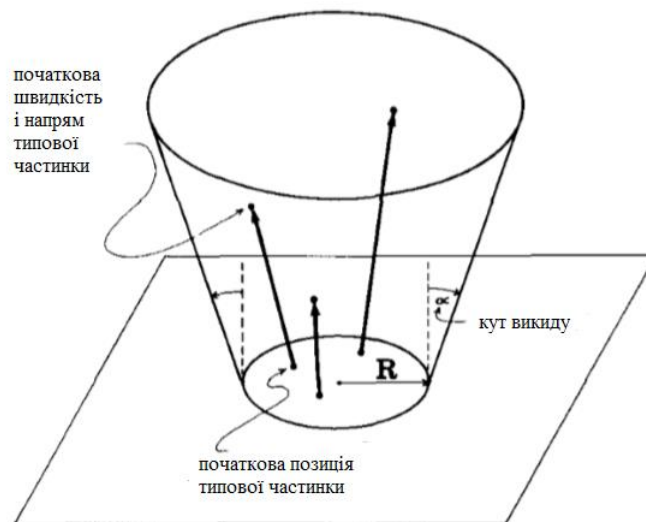


Рисунок 2.4 - Форма генерації системи частинок

2.5 Динаміка частинок

Окремі частки в системі частинок рухаються в тривимірному просторі, а також з часом змінюються за кольором, прозорості та розміром.

Перенести частку з одного кадру в інший - це просте питання про додавання вектора швидкості до вектору положення. Щоб додати додаткову складність, система

частинок також використовує коефіцієнт прискорення для зміни швидкості його часток від кадру до кадру.

2.6 Візуалізація частинок

Після того, як параметри стану та зовнішнього вигляду всіх частинок були розраховані для кадру, алгоритм рендерінга створює зображення. Загальна проблема рендерінга частинок настільки ж складна, як і рендерінг об'єктів, що складаються з більш поширених графічних примітивів, таких як полігони та криволінійні поверхні. Частинки можуть приховувати інші частинки, які знаходяться за ними в глибині екрана. Вони можуть бути прозорими і можуть відкидати тіні на інші частинки. Крім того, частинки можуть співіснувати в сцені з об'єктами, змодельованими примітивами на поверхні, і ці об'єкти можуть перетинатися з частинками.

В нашій існуючій системі два припущення дозволяють спростити алгоритм рендерінга. По-перше, ми припускаємо, що системи частинок не перетинаються з іншими примітивами поверхневого моделювання, і тому нашого алгоритму рендерінга необхідно обробляти частки. Об'єкти, змодельовані з використанням інших методів, компонуються разом з об'єктами системи частинок в стадії пост-рендерінга. Щоб система частинок перетиналася або перебувала позаду інших об'єктів, система рендерінга розділила зображення системи частинок на підзображення на основі площин відсікання, визначених у просторі координат моделі. Ці підзображення потім об'єднуються з іншими зображеннями на стадії компонування.

Інше спрощуюче припущення, зроблене в нашій поточної системі рендерінга, полягає в тому, що кожна частка може відображатися як точкове джерело світла. З урахуванням цього припущення, визначення прихованих поверхонь вже не є

проблемою. Кожна частка додає трохи світла до пікселів, які вона покриває. Частка за одною часткою НЕ затінена, а скоріше додає більше світла до покритих пікселей. Кількість доданого світла і його колір залежать від прозорості і кольору частки. Перетворення перегляду, розмір частки і її форма визначають, які пікселі покриті. Всі форми частинок намальовані згладжуванням, щоб запобігти тимчасовим накладенням і стробіюванням. Світло може бути додане до пікселя багатьма частинками, тому алгоритм рендерінга захоплює окремі інтенсивності червоного, зеленого і синього при максимальному значенні інтенсивності буфера кадру, а не дозволяє їх обгортати.

За допомогою цього алгоритму і припущень не вимагається сортування частинок. Вони відображаються в буфер кадру в будь-якому порядку, в якому вони створюються. Тіні не становлять проблему, тому що частинки не відображають, а випромінюють світло.

2.7 Ієрархія частинок

Система має механізм, який підтримує формування і контроль ієрархії систем частинок. Створюється система частинок, в якій частинки самі є системами частинок. Коли вихідна система частинок трансформується, трансформуються всі її потокові системи частинок і їх частки. Середній колір вихідної частинки і її дисперсія використовуються для вибору середнього кольору і дисперсії систем частинок потомства з використанням тих же рівнянь, які були представлені раніше. Число нових систем частинок, що генеруються в кадрі, засноване на швидкості генерації часток джерела. Інші параметри джерела аналогічно впливають на параметри його похідних. Структура даних, використовувана для подання ієрархії, являє собою дерево.

Ієрархію можна використовувати для глобального управління складним нечітким об'єктом, що складається з багатьох систем частинок. Наприклад, хмара може складатися з багатьох систем частинок, кожна з яких представляє собою хвильову область частинок води. Головна система частинок групує всі похідні разом і управляє глобальним рухом і зовнішнім виглядом хмари під впливом вітру і місцевості [7].

2.8 Методи

Гідродинаміка згладжених частинок (англ. Smoothed Particle Hydrodynamics, SPH) - обчислювальний метод для моделювання динаміки рідини і газів. Використовується в багатьох областях досліджень, включаючи астрофізику, балістику, вулканології і океанографії. Метод гідродинаміки згладжених частинок є безсітковим (англ. Mesh-free) лагранжевим методом (тобто координати рухаються разом з рідиною), і роздільна здатність методу може бути легко відрегульована щодо змінних, таких як щільність.

Метод SPH працює шляхом ділення рідини на дискретні елементи, звані частками. Ці частинки мають просторове відстань (відоме як «довжина згладжування», зазвичай надається в рівняннях як h), на якому їх властивості «згладжуються» функцією ядра. Це означає, що будь-яка фізична величина будь-якої частинки може бути отримана шляхом підсумовування відповідних величин всіх частинок які знаходяться в межах двох згладжених довжин. Наприклад, температура в точці R залежить від температури всіх частинок на відстані $2h$ від R .

Вплив кожної частки на властивості оцінюється відповідно до її щільністю і відстанню до цікавить частки. Математично, це описується функцією ядра (позначається W). В якості опції ядра часто використовують функцію Гаусса (функція

нормального розподілу) або кубічний сплайн. Остання функція дорівнює нулю для частинок знаходяться далі ніж дві згладжені довжини (на відміну від функції Гаусса, де є невеликий вплив на будь-якому кінцевому відстані). Це дозволяє економити обчислювальні ресурси, виключаючи відносно мала вплив віддалених частинок.

Значення будь-якого фізичного величини A в точці r , задається формулою:

$$A(r) = \sum_j m_j \frac{A_j}{p_j} W(|r - r_j|, h),$$

де m_j - маса частинки j , A_j - значення величини A для частинки j , p_j - щільність пов'язана з часткою j , і W - функція ядра згадана вище.

Наприклад, щільність частки i (p_i) Може бути виражена як:

$$p_i = p(r_i) = \sum_j m_j \frac{p_j}{p_j} W(|r_i - r_j|, h) = \sum_j m_j W(|r_i - r_j|, h),$$

де підсумовування j включає всі частинки в симуляції.

Аналогічно, просторова похідна кількості може бути отримана за допомогою інтегрування частинами для зміщення оператора набла (∇) від фізичної величини до функції ядра:

$$\nabla A(r) = \sum_j m_j \frac{A_j}{p_j} \nabla W(|r - r_j|, h).$$

Хоча розмір довжини згладжування може бути фіксованим і в просторі, і в часі, це не дозволяє використовувати SPH в повному обсязі. Призначаючи кожній частинці її власну довжину згладжування і дозволяючи їй змінюватися з часом, роздільна

здатність симуляції може автоматично підлаштовувати себе до локальних умов. Наприклад, в дуже щільній області, де багато частинок розташовані близько одна до одної, довжина згладжування може бути зроблена відносно короткою, приводячи до високого просторового розширення. І навпаки, в областях з малою щільністю, де частки розміщені далеко одна від одної і роздільна здатність низька, довжина згладжування може бути збільшена, оптимізуючи обчислення для даної області. Об'єднана з рівнянням стану і інтегратором, гідродинаміка згладжених частинок може ефективно симулювати гідродинамічні потоки. Однак традиційна штучна формулювання в'язкості, яка використовується в гідродинаміки згладжених частинок, має тенденцію «змащувати» (англ. Smear out) ударні хвилі і контактні розриви в набагато більшому ступені, ніж сучасні засновані на сітках методи.

Адаптивність гідродинаміки згладжених частинок, заснована на Лагранжевому підході, аналогічна адаптивності сучасних сіткових кодів із застосуванням адаптивних сіток уточнення адаптивної мережі (англ.) (Англ. Adaptive mesh refinement), хоча в останньому випадку можна подрібнити сітку згідно будь-яким критерієм, вибраним користувачем. Оскільки гідродинаміка згладжених частинок Лагранжева за своєю природою, то вона обмежена в параметрах подрібнення, використовуючи тільки щільність.

Часто в астрофізиці необхідно змодельовати гравітацію на додаток до гідродинаміки. Заснована на частинках «природа» SPH робить її ідеальним вибором для об'єднання з обробником гравітації, який заснований на частинках.

Гідродинаміка згладжених частинок все більш часто використовується для моделювання руху рідин. Це відбувається через деяких переваг методу SPH в порівнянні з традиційними заснованими на сітці методиками. По-перше, SPH гарантує збереження маси без додаткових обчислень, так як частинки самі по собі представляють масу. По-друге, SPH обчислює тиск від впливу сусідніх частинок, також мають масу, а не вирішує систему лінійних рівнянь. Нарешті, на відміну від

заснованих на сітці методик, які повинні простежувати кордону рідини, SPH створює вільну поверхню для безпосередньо двофазних взаємодіючих рідин, так як частинки являють більш щільну рідину (зазвичай воду), а вільний простір являє легшу рідину (зазвичай повітря). З цих причин завдяки SPH можливо моделювати рух рідини в режимі реального часу. Однак, і SPH, і засновані на сітці методики все ще потребують в візуалізованій вільній поверхневій геометрії і використовують полігонізаційні методики, такі як metaballs, marching cubes, point splatting або «килимову» візуалізацію («carpet» visualization). Для газу більш доречно використовувати безпосередньо функцію ядра, щоб зробити рендеринг щільності газу (наприклад, як зроблено в пакеті візуалізації «SPLASH»).

Єдиний недолік SPH в порівнянні з заснованими на сітці методиками полягає в тому, що необхідна велика кількість частинок для створення симуляції з еквівалентної роздільною здатністю. У типовій реалізації заснованих на сітці методик і SPH, багато вокселів або частинок будуть перебувати під поверхнею води, в глибині водяного обсягу, і ніколи не будуть візуалізовані. Однак точність може бути значно збільшена зі складними заснованими на сітці методиками, особливо з тими, які використовуються спільно з методами частинок (такими, як набори рівнів частинок).

Для некритичних додатків, таких як комп'ютерні ігри і кінофільми, продуктивність і візуальний реалізм набагато більш значущі, ніж обчислювальна точність. Muller і інші використовували SPH для симуляції води, яка тече в склянку. При цьому використовувалося кілька тисяч частинок, а частота кадрів становила близько 5 кадрів / сек. Kipfer і Westermann (Technical University at Munich, Germany) використовували SPH для симуляції річки. Такахіро Харада (англ. Takahiro Harada) та інші використовували сучасні графічні процесори GeForce 8800 GTX для симуляції 49 153 частинок зі швидкістю 17 кадрів / сек [1].

Програмне забезпечення по емуляції SPH:

- RealFlow - офлайновий фізичний двигун, призначений для використання в індустрії комп'ютерної графіки, анімації та спецефектів і використовує SPH.
- FLUIDS v. 3 - проста, вільна (ліцензія zlib) тривимірна реалізація SPH в режимі реального часу для рідин, написана на C++ і використовує для розрахунків CPU і GPU.
- GADGET (англ.) - вільно доступний код для космологічних симуляцій N-body / SPH.
- SPLASH - вільно доступний інструмент для візуалізації SPH-симуляцій.
- SPHysics - реалізація SPH з відкритим вихідним кодом, написана на Фортране.
- Physics Abstraction Layer (PAL) (англ.) - вільна система абстракції, яка підтримує фізичні движки реального часу з підтримкою SPH.
- Pasimodo - програмний пакет для заснованих на частинках методів симуляції, включаючи SPH.

3.9 Оптимізація

Одна з найсерйозніших завдань художника візуальних ефектів при використанні ресурсів в грі - це оптимізація. Розглянемо для прикладу ігровий двигун Unreal. Написаний на мові C++, двигун дозволяє створювати ігри для більшості операційних систем і платформ. Найважливіше для нас слово - це «перемалювання» (overdraw). Зазвичай ігрові двигуни рендерять кінцеве зображення за кілька проходів (albedo, roughness, metalness ...) і в кожному кадрі, в якому на екрані знаходиться частинка, вона впливає на конвеєр рендеринга. Залежно від типу рендеринга цей вплив буває різним. Наприклад, якщо у нас на екрані є дві прозорі (translucent) частки, що накладаються один на одного, то витрати на рендеринг цих пікселів подвоюються. Це особливо погано для продуктивності, коли частинки повноекранні, тому що двигуну потрібно виконувати рендеринг X раз 1920×1080 пікселів (в разі Full HD).

Найчастіше елементи не найскладніші: частинки з сіток мають мало трикутників, шейдери не надто складні, частки висвітлюються набагато більш «дешевими» способами, ніж персонажі або оточення, деякі частинки взагалі не висвітлюються, і т.д. Але навіть з урахуванням цього частинки можуть бути дуже витратними під час виконання, і це залежить від способу їх рендерінга. Якщо спростити, то є три основних способи рендерінга частинок (в Unreal вони називаються Blend Mode):

Непрозорий (Opaque): найдешевший, але його рідко використовують, якщо тільки ми не говоримо про частки з сіток. Це найдешевший метод, тому що частинки, що перекривають інші частинки, утримують їх. Тому двигуну простіше відкидати пікселі.

З альфа-маскою (Alpha masked): також він називається альфа-тестом (alpha test). У ньому у нас є значення прозорості 0 і 1. Тому пікселі бувають тільки видимими і невидимими, без напівпрозорості. Це найпростіший спосіб отримання прозорості, принаймні, в консольних іграх. Непрозорі (opaque) пікселі відсікають пікселі під ними, так само, як в описаному вище способі з непрозорістю.

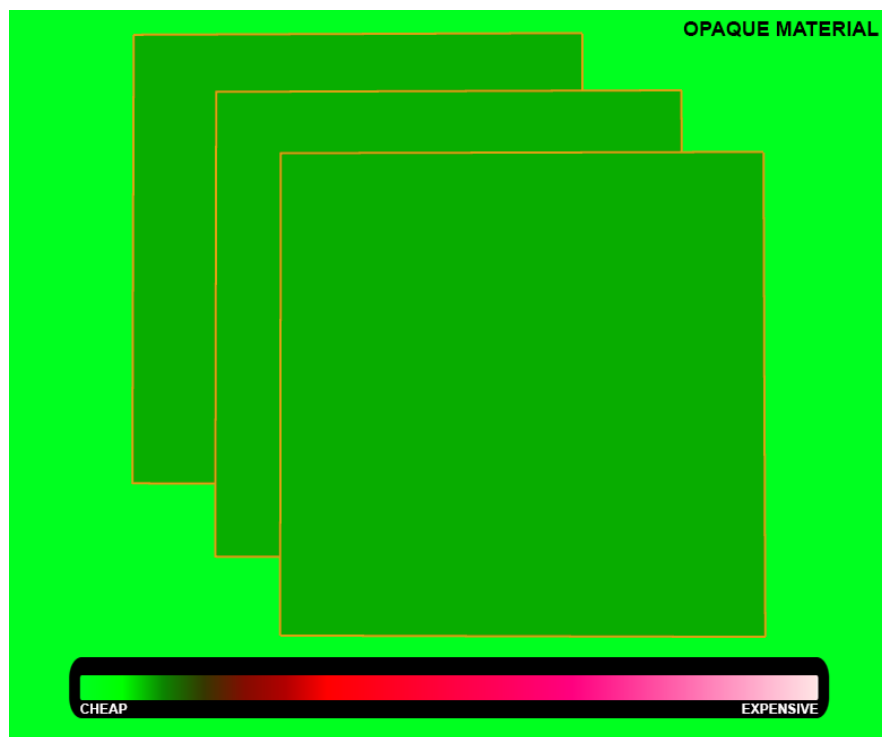


Рисунок 2.4 Спосіб рендерінгу Оpaque

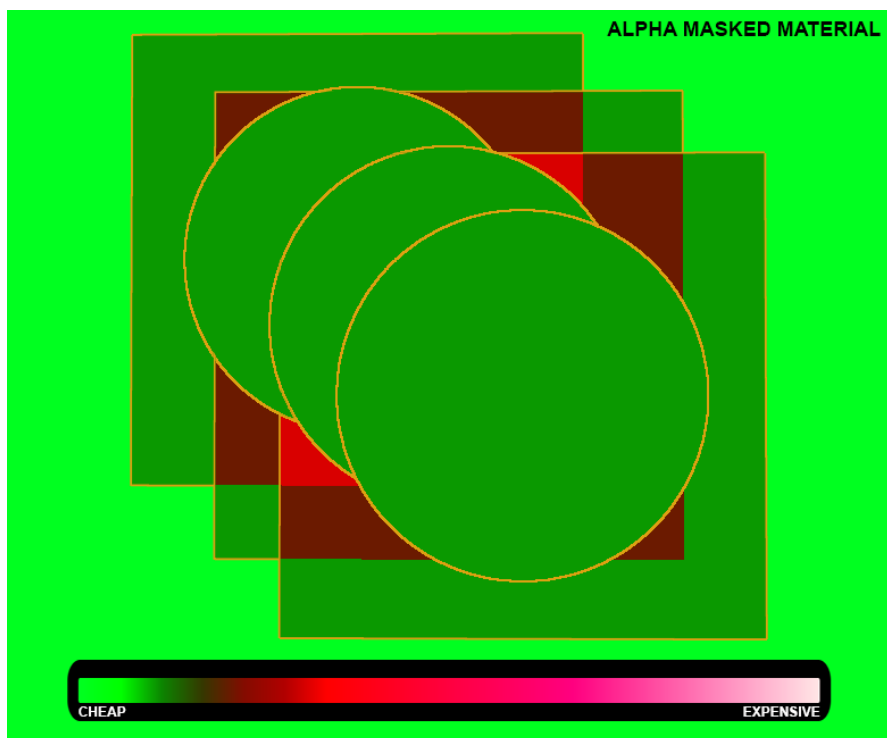


Рисунок 2.5 Спосбі рендерінгу з використанням альфа маски

Прозорий (Translucency): також він називається альфа-змішуванням (alpha blend). В цьому випадку у нас немає ніякого відсікання. Це самий витратний спосіб рендеринга частинок. Найбільшою проблемою для продуктивності є те, що движкам складно впорядкувати рендеринг напівпрозорих пікселів, тому що вони зазвичай не зберігають глибину по Z (Z depth) (а непрозорі і з альфа-маскою зберігають). Саме тому ви найчастіше бачите гальмування в іграх (туман, пил, вікна і т.д.).

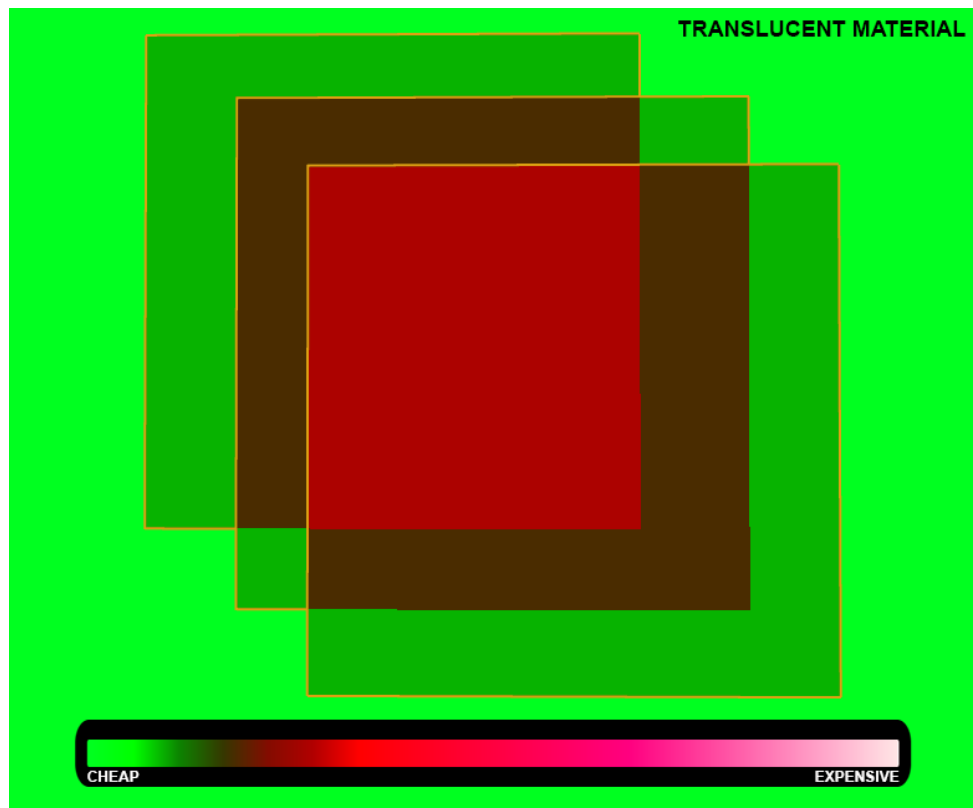


Рисунок 2.6 Спосіб рендерінгу Translucency

Крім того, є два способи освітлення:

Неосвітлені (Unlit): частинки взагалі не висвітлюються. Все освітлення потрібно записати в текстуру або виконувати через шейдер (а це не зовсім точний спосіб).

Освітлені (Lit): частинки отримують освітлення, схоже з освітленням оточення і пропсов. Цей спосіб тривіальний для частинок без прозорості / з альфа-маскою, тому

що вони зберігають інформацію про глибину по Z . Однак за замовчуванням просвічуються частки не зберігають цю інформацію, тому движок не знає, де ці частинки знаходяться в світі. Додавання просвічується часткам освітлення додає ще один рівень складності до рендерінгу кожної частки.

Додавання освітленості до просвічується часткам дозволяє напівпрозорим часткам зберігати інформацію про глибину по Z . Саме по собі це дуже затратно, але сучасні графічні карти при акуратному використанні способу можуть впоратися з цим.

Для оптимізації спеціальних ефектів необхідно:

- Використовувати непрозорі частки. Деякі або всі об'єкти з частинками можуть бути непрозорими, з вирізанням альфа. Для деяких частинок, таких як шрапнель, камені або аналогічні, що використовують легкі частинки геометрії замість спрайтів з альфа-кордонами.
- Зниження загальної кількості частинок. Апаратні лічильники на графічній карті використовуються щоб підрахувати, скільки пікселів частинок було відображено, щоб припинити випромінювати або малювати частки при проходженні певної межі (який можна встановити динамічно).

Так само знижують можливі проблеми зменшення необхідності сортування, використовуючи адитивно- або відємно-змішані частинки. Наприклад, адитивні частки можуть бути намальовані в будь-якому порядку, тому ми можемо сортувати їх, наприклад, для зменшення змін стану замість сортування по глибині (що, ймовірно, необхідно для нормальних альфа-змішаних часток).

Кількість частинок на екрані безпосередньо впливає на продуктивність: чим їх більше, тим вона гірше. Так само дуже важливий розмір частки. GPU доводиться дуже багато прораховувати накладаються один на одного напівпрозорі шари. І чим їх більше, тим складніше прорахунок. І якщо їх площа близька до загальної площі екрану, то розрахунків доводиться робити більше, ніж для маленьких частинок. Адже

потрібно розрахувати, як буде виглядати той, що знаходиться за цими великими напівпрозорими частинками. Тому важливим є використання їх дуже обережно і необхідно звести їх кількість і час життя до мінімуму. Іншими словами, зробити дим на підлогу екрана краще з 200 невеликих частинок, ніж з 40-50 великих. Візуально важко побачити різницю, однак це допоможе вирішити питання продуктивності.

Чим швидше «вмирають» частинки, тим їх менше на екрані одночасно і тим краще продуктивність. Виняток становлять дуже повільні ефекти на зразок стовпа диму вдалині, де велику відстань до ефекту підкреслюється його неквапливістю. Це правило так само варто пам'ятати при створенні ефектів. Чим повільніше рухається ефект, тим далі від глядача він сприймається. І навпаки відповідно. Але і в ньому, звичайно, бувають винятки.

Висновки до розділу

1. Представлено метод системи частинок для моделювання нерегулярних об'єктів таких як хмари, дим, вода та вогонь, в якому математично кожна частка представляється як матеріальна точка з додатковими атрибутами, такими як швидкість, колір, орієнтація в просторі, кутова швидкість, і т.п.
2. Показано, для моделювання спеціальних ефектів актуальною є проблема оптимізації для досягнення прийняттого відношення реалістичності ефекту до частки зайнятих ним апаратних потужностей. Для вирішення даної задачі необхідно враховувати вплив типу, кількість та способи рендерінгу частинок.

3 СТВОРЕННЯ СПЕЦІАЛЬНОГО ВІЗУАЛЬНОГО ЕФЕКТУ

3.1 Створення ефекту полум'я у програмі *Magic Particles*

Для дослідження створення ефекту полум'я за допомогою системи частинок використана програма *Magic Particles* - редактор візуальних ефектів. Програма призначена для швидкого створення барвистих і динамічних спецефектів.

Для створення реалістичного вогню дуже важливі текстури частинок. *Magic Particles* працює з форматом файлів PSD(Photoshop Document), що є дуже зручним. Для створення використано аддитивний блендинг (режим накладання в *Photoshop*).

При такому режимі змішування, чорний колір є абсолютно прозорий, а інші кольори висвітлюють картинку під ними. Для створення частинок вогню намальовані текстури 256x256 на чорному фоні(див. Рис.3.1).



Рисунок 3.1 – Текстури, намальовані у Photoshop

Використано тонкий пензлик з м'якими краями, а потім розмазано краї «пальцем», за необхідністю продублювавши шари та розмиваючи їх **Gaussian Blur**(див. Рис.3.2)

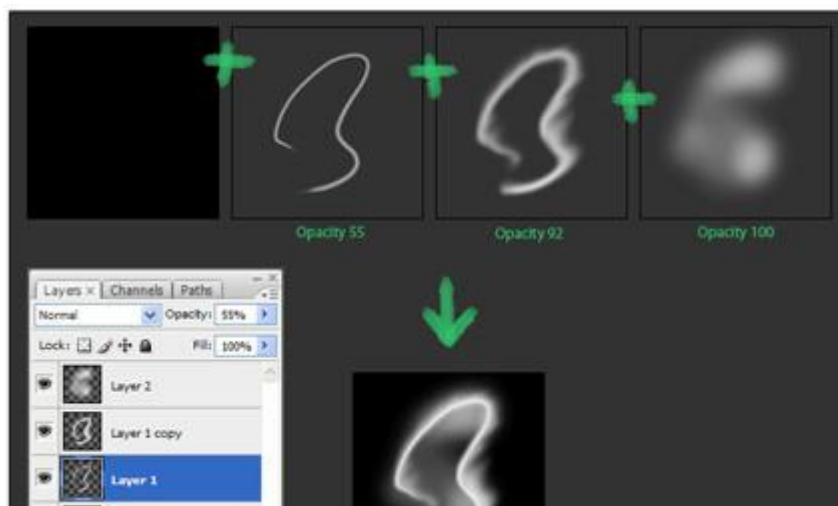


Рисунок 3.2 – Процес створення текстури

Після створення текстури переходимо до програми *Magic Particles*. На панелі кнопок вибрано кнопку «створити емітер», а далі «створити тип частинок всередині емітера»(див. Рис.3.3).

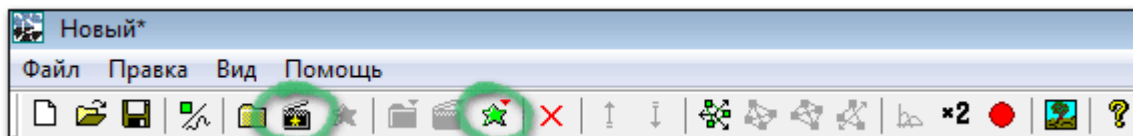


Рисунок 3.3 – Верхня панель програми *Magic Particles*.

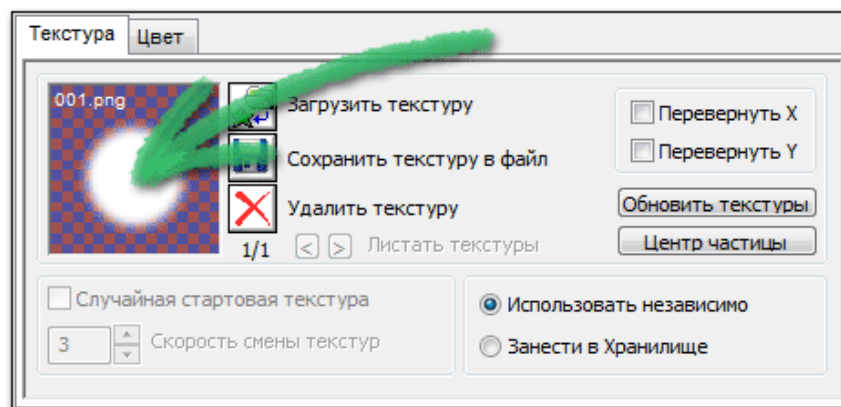


Рисунок 3.4 – Меню текстури

У вікні перегляду відображена іконка емітера, в квадратній рамці. Саме в цьому вікні завантажуються всі текстури(див. Рис.3.4).

На вкладці кольору за допомогою параметру **Інтенсивність** активізується аддитивний блендинг, при якому змішування кольору додається висвітлюючи картинку перед собою, а чорний екран є абсолютно прозорим – частинки світяться.

Налаштування кольору представлено на Рис.3.5.

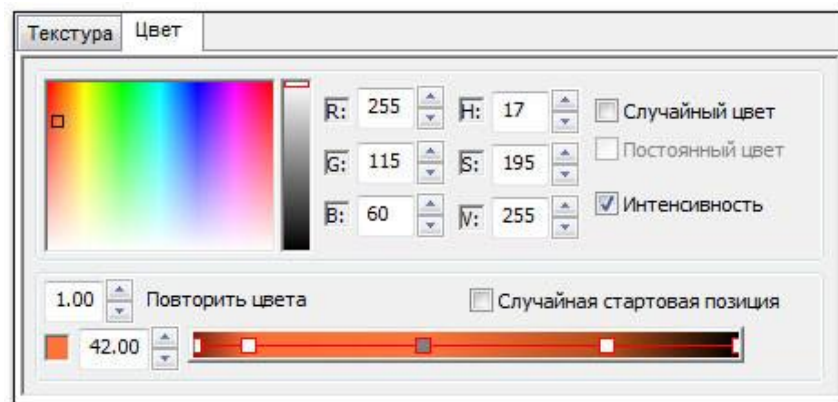


Рис.3.5 – Вікно налаштування параметрів кольору

Останній маркер чорний для плавного зникнення частинки.

На вкладці **Текстура** параметр **Швидкість зміни текстури** вибрана нулю та ввімкнутий параметр **Випадкова початкова текстура**. Таким чином для кожної частинки вибирається випадкова текстура, що незмінна до кінця її тривалості.

Для анімації частинок на вкладці **Дерево емітерів** змінено тип емітера за точки на лінію. Це джерело нашої системи. Ця лінія має два параметри: синій графік – кут нахилу, зелений – довжина лінії. Початкові параметри(кут – 0, довжина – 100) задовольняють наші умови.

Наступний графік **Напрямок емітера, градус** змінює кут випускання частинок. Для нашого випадку вибрані кути 36° і 144° , для вертикального випускання частинок.



Рисунок 3.6 – Налаштування графіків

На вкладці **Додатково** зібрані такі параметри як **Життя**, **Кількість**, **Розмір**, **Швидкість**. Вони взаємозв'язані – високий параметр **Швидкість** збільшить темп руху частинок і одночасно подовжить полум'я, але, знизив значення **Життя**, можливо зменшити висоту.

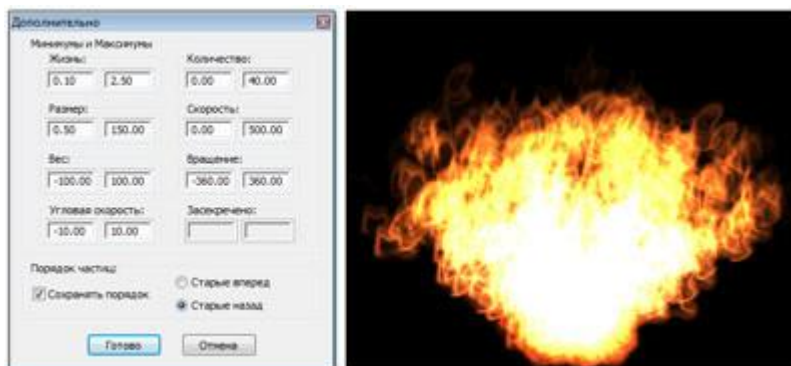


Рисунок 3.7 – Налаштування додаткових параметрів

Зменшивши значення на графіку **Ступінь непрозорості(коефіцієнт)** зменшиться щільність та яскравість полум'я.

Використовуючи параметр **Вага** з від'ємними значеннями змусить частинки плавно підійматися.

В *Magic Particles* є два типи графіків:

- Звичайні(сірі іконки)
- Коефіцієнти (червоні іконки)

Звичайні графіки регулюють властивості частинки протягом існування ефекту(вісь Х- життя ефекту), а **коефіцієнти**, міняють частинку протягом її життя (вісь Х- життя частинки)(див Рис.3.8 – 3.9).

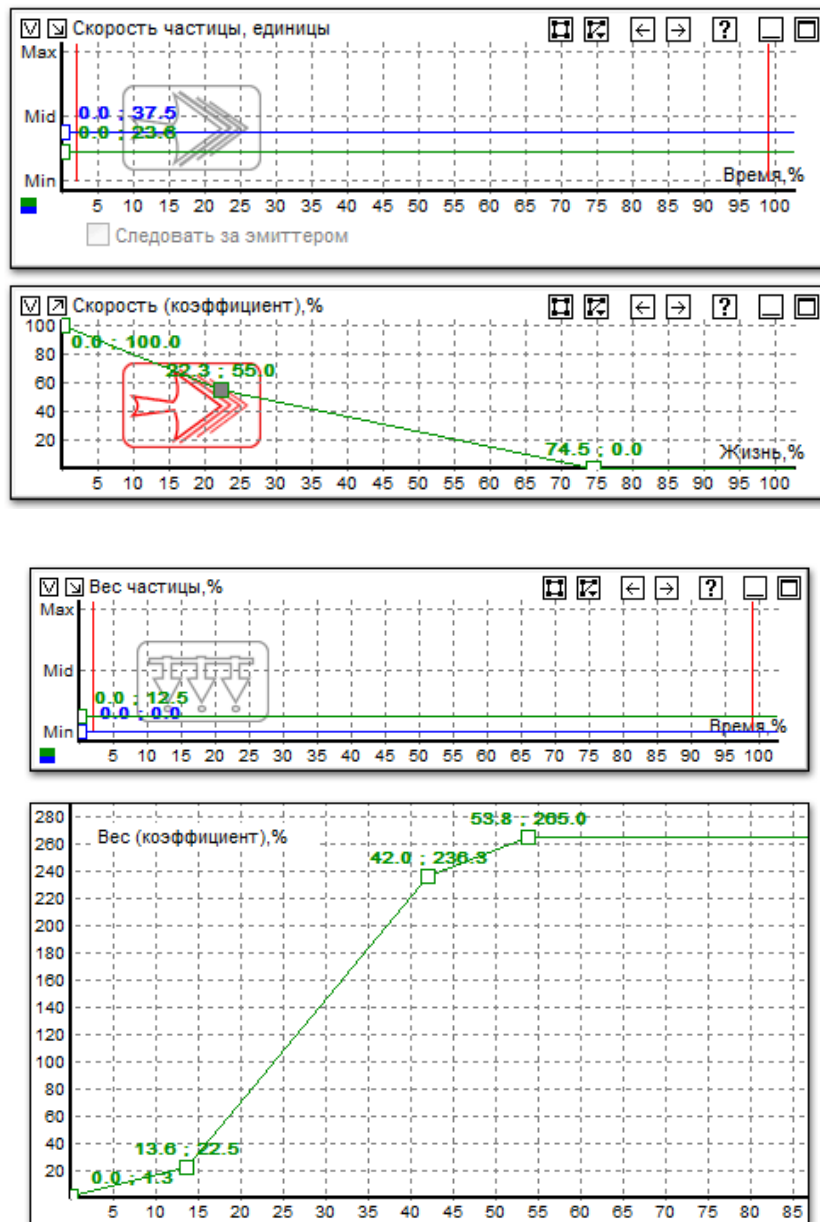


Рисунок 3.8 – Графіки налаштування необхідного руху частинок

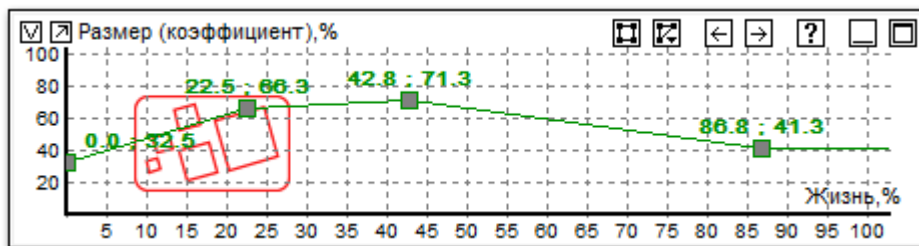


Рисунок 3.9 – Налаштування форми полум'я

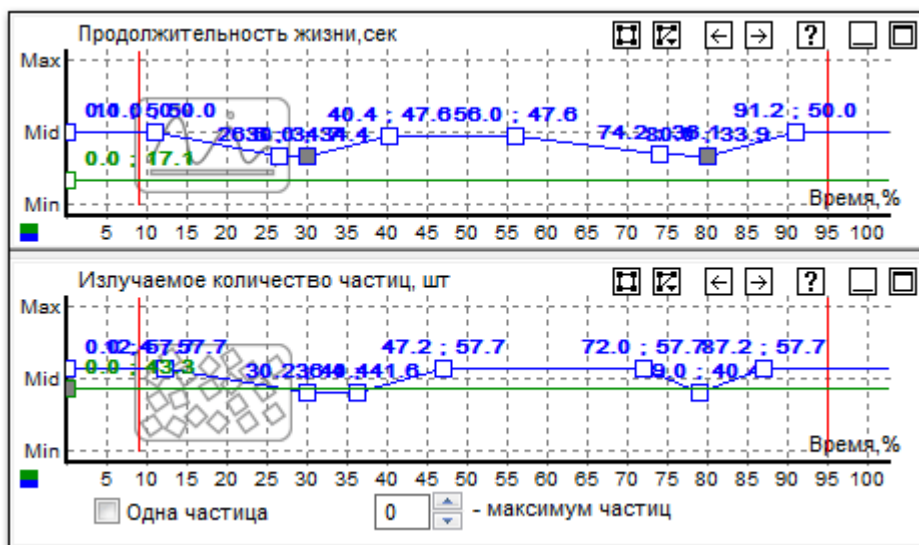


Рисунок 3.10 – Налаштування кількості частинок та часу їх існування

Графіки з двома лініями дають можливість задавати випадкові значення у вибраному діапазоні. Тобто кожній створюваній частинці будуть присвоєні різні значення параметра.

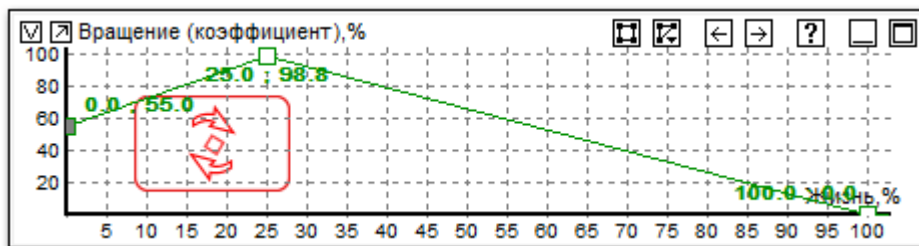


Рисунок 3.11 – Налаштування обертання для збільшення реалістичності



Рисунок 3.12 – Результат моделювання

3.2 Створення ефекту полум'я у програмі *After Effects* на основі системи частинок - CC Particle World

Інтерфейс ефекту CC Particle World зображено на Рис.3.14.

За допомогою властивостей ефекту здійснюється управління системою частинок:

- Grid & Guides – налаштування сітки.
- Birth Rate – кількість частинок, що з'являються на екрані.
- Longevity визначає тривалість існування частинки.
- Producer. Position - визначення точки-джерела системи частинок, Radius задає виліт частинки не з однієї точки, а з осі.
- Physics – розділ управління поведінкою частинок.
 - Пункт Animation визначає характер руху частинок: Explosive - вибух, Directions Axis – напрямок осі, Cone Axis – конічна вісь, Viscous -

в'язкість, Twirl - обертання, Vortex - вихор, Fire – вогонь, Jet Sideways – реактивний, Fractal – фрактал (див. Рис.3.15).

- Velocity – швидкість переміщення частинок.
- Gravity – вага частинок у просторі.
- Directions – зміна напрямку випромінювання.
- Particle визначає форму частинок:
 - Particle Type - тип частинки: Line - лінія, Star - зірка, Sphere - сфера, Bubble - бульбашка, Polygon - полігон, Tetrahedron - тетраедр, Cub - куб, Lens – лінза, Texture – текстура та їх модифікації.
 - Birth Size/Color – розмір і колір частинки при ініціалізації.
 - Death Size/Color – розмір і колір частинки на кінці шляху.
 - Transfer Mode – кольоровий режим при накладанні.

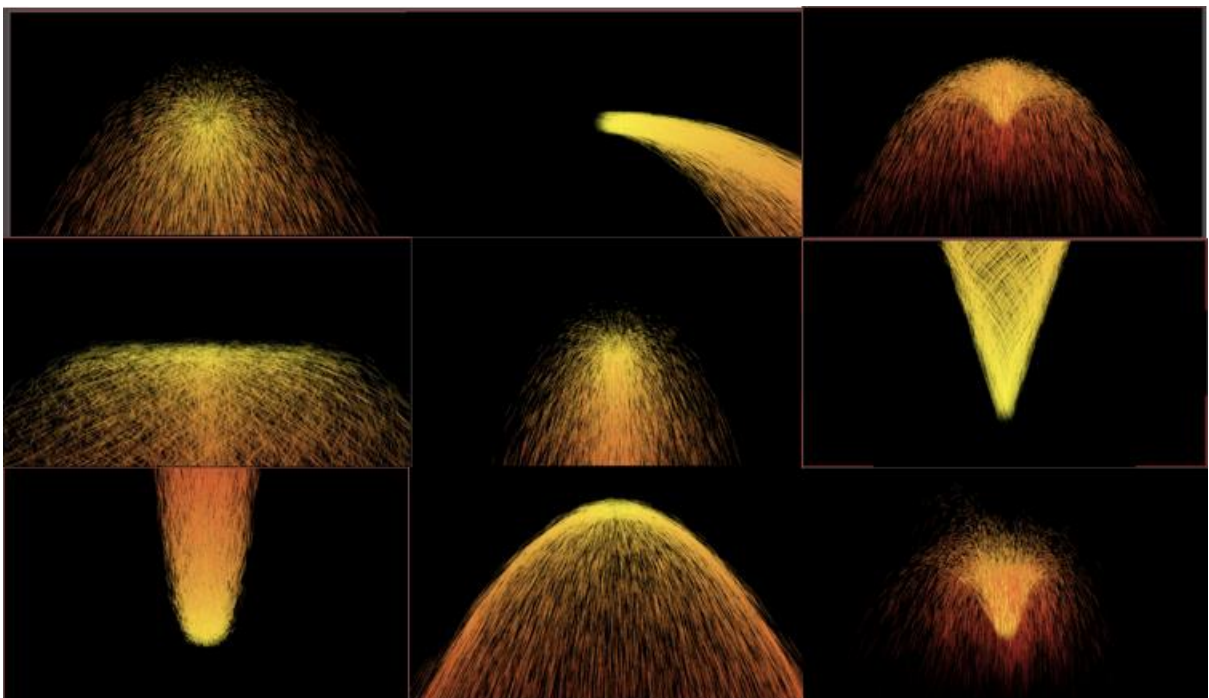


Рисунок 3.15 – параметр Animation – доступні характери руху частинок

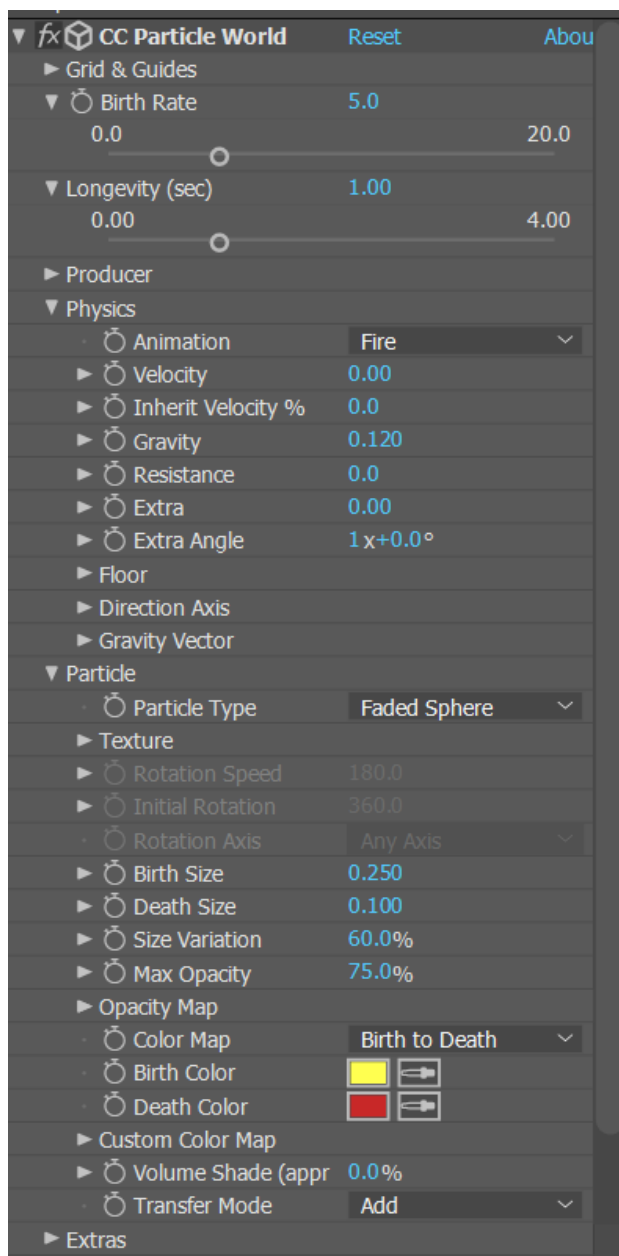


Рисунок 3.14 – Інтерфейс ефекту CC Particle World в програмному середовищі Adobe After Effects CC 2017

3.2.1 Створення ефекту полум'я на основі системи частинок - CC Particle World з використанням типу частинки – Затінена сфера

Для створення полум'я на суцільний шар застосовано ефект CC Particle World (див. Рис.3.15).

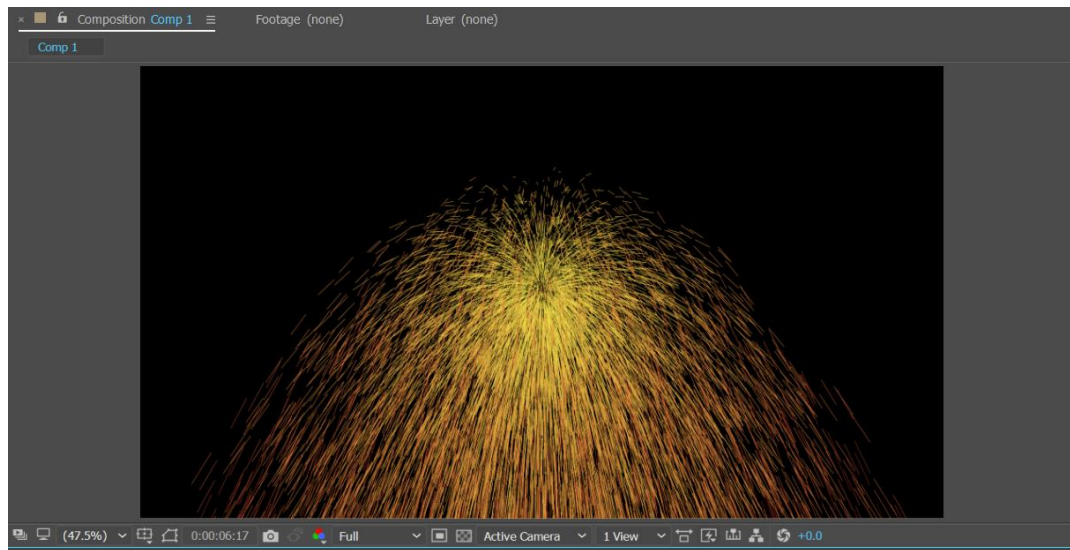


Рисунок 3.16 – Ефект CC Particle World з початковими параметрами

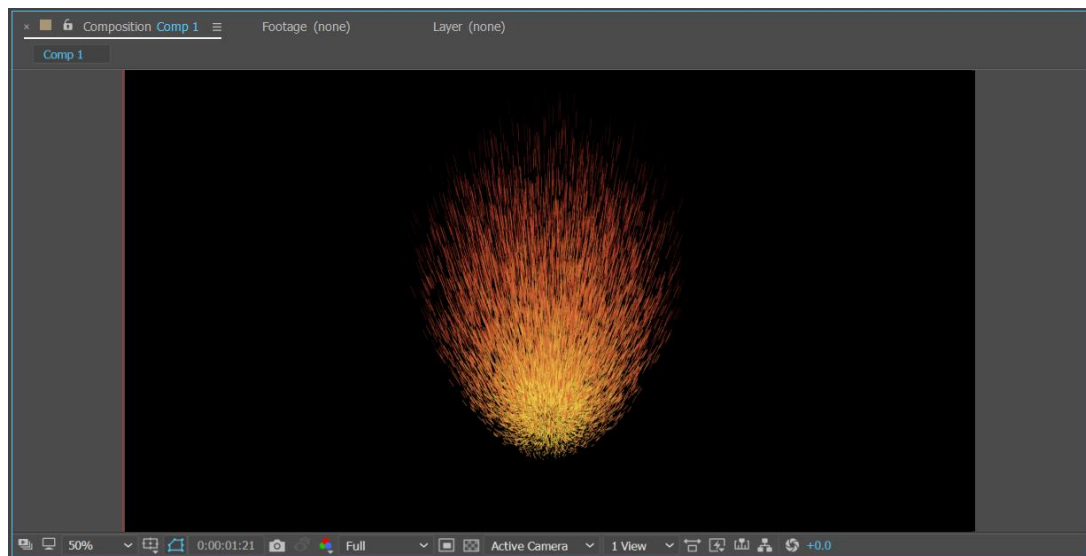


Рисунок 3.17 - Зміна характеру руху частинок на Fire та зменшення маси частинки для більш плавного поширення

Далі представлена робота з типом частинки – Затінена сфера (Shaded Sphere)(див. Рис.3.18). Зниження швидкості (параметр Velocity) і маси частинки (Gravity), робить їх переміщення більш плавним – більш схожим на язики полум'я (див. Рис.3.19).

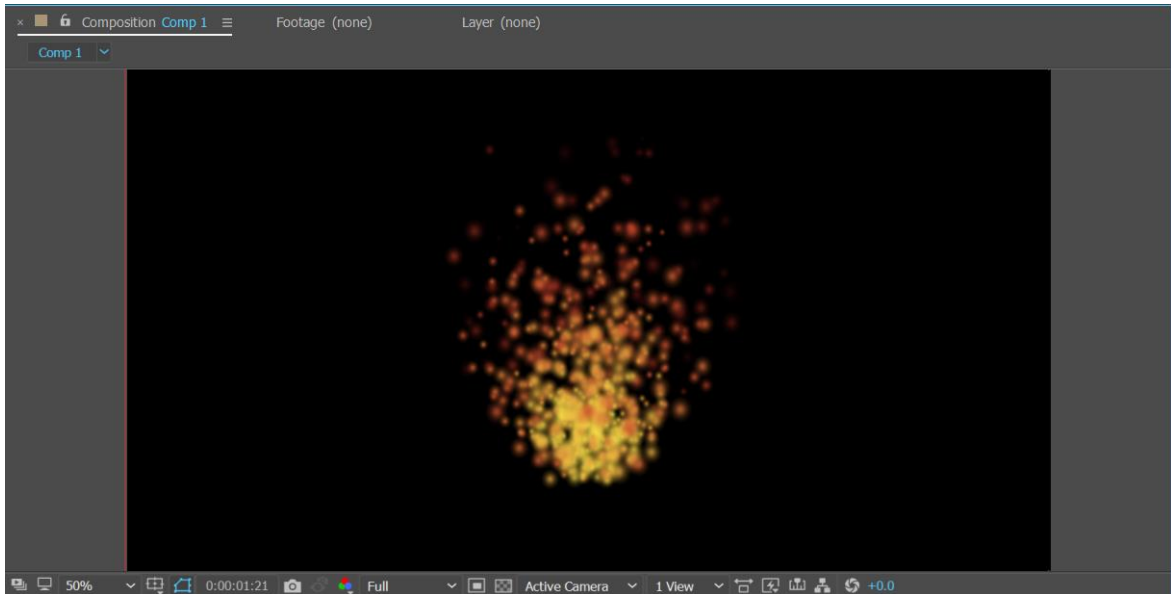


Рисунок 3.18 – Particle Type – Shaded Sphere

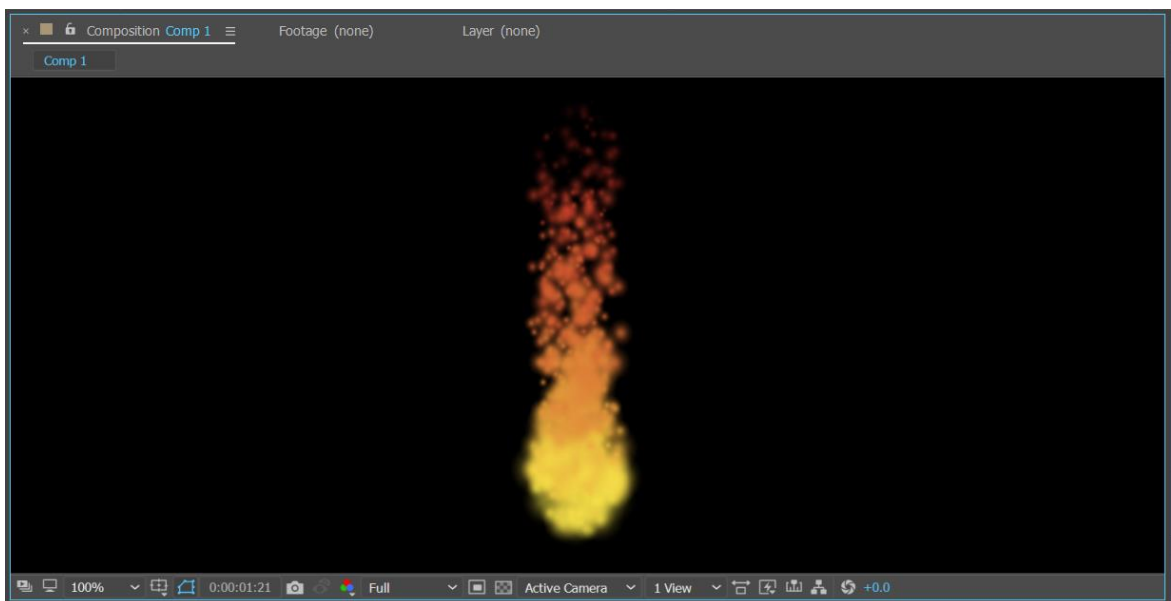


Рисунок 3.19 – Робота з параметрами Velocity та Gravity

Transfer Mode змінений на Add для більш яскравого свічення у місцях перекриття частинок. На цьому робота з ефектом CC Particle World закінчена і необхідна допомога наступних ефектів AE:

- Directional Blur, Vector Blur – ефекти розмиття об'єкту (див. Рис.3.20).
- Glow – ефект свічення (див. Рис.3.21).

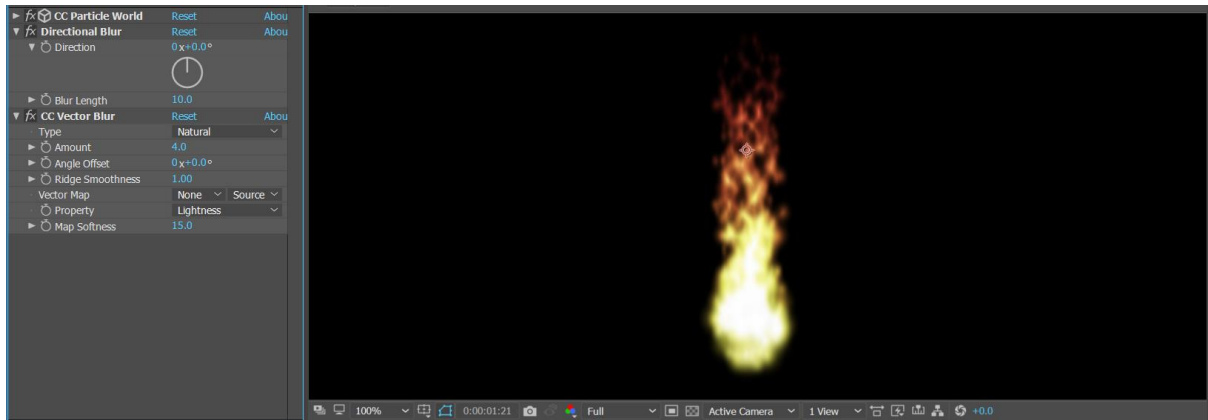


Рисунок 3.20 – Параметри та результат накладання ефектів Directional та Vector Blur

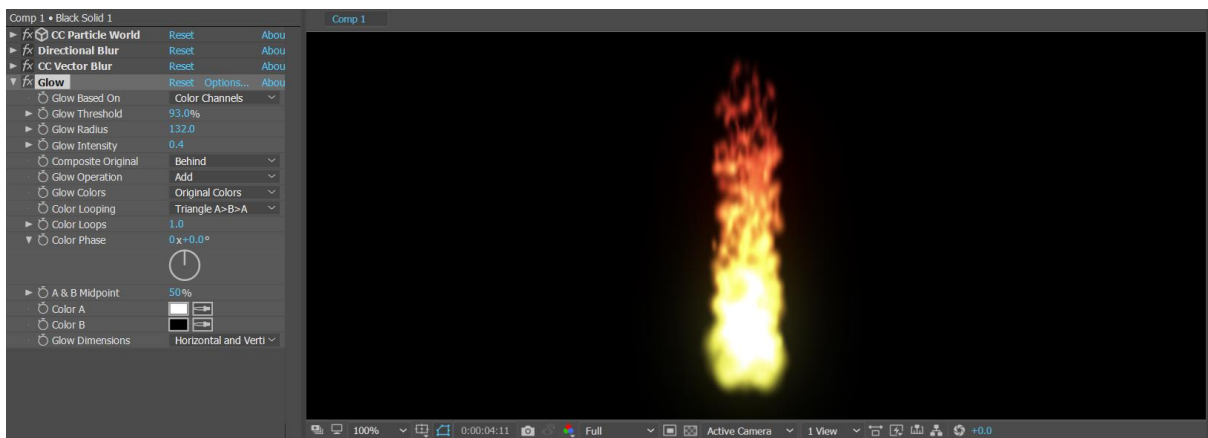


Рисунок 3.21 - Параметри та результат накладання ефекту Glow

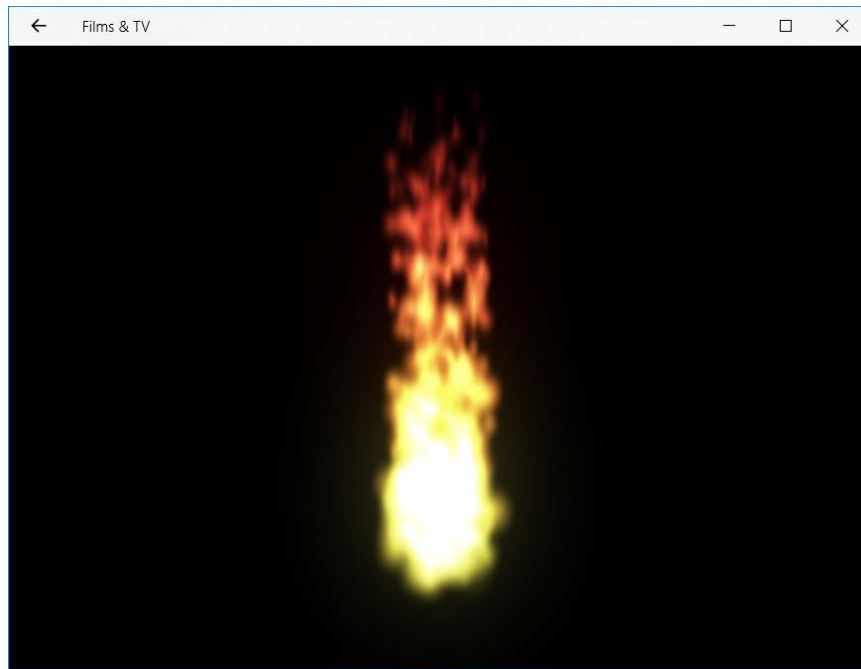


Рисунок 3.22 – Результат рендерінгу

3.2.2 Створення ефекту полум'я на основі системи частинок - CC Particle World з використанням типу частинки – Лінза-бульбашка

Замість типу частинки Затінена сфера (Shaded Sphere) у розділі вибору типу частинки вибрано тип Лінза-бульбашка (Lens Bubble) (див. Рис.3.23) та видалено усі допоміжні ефекти з попереднього проекту.

Допоміжні ефекти:

- Fast Blur – розмиття вздовж вертикальної осі.
- Turbulent Displace – додавання вихрового зміщення та еволюції цього зміщення (див. Рис.3.25).
- Colorama – додавання кольору, використовуючи альфа канал (див. Рис.3.26).
- Sharpen – збільшення чіткості та Glow – ефект свічення (див. Рис.3.27).

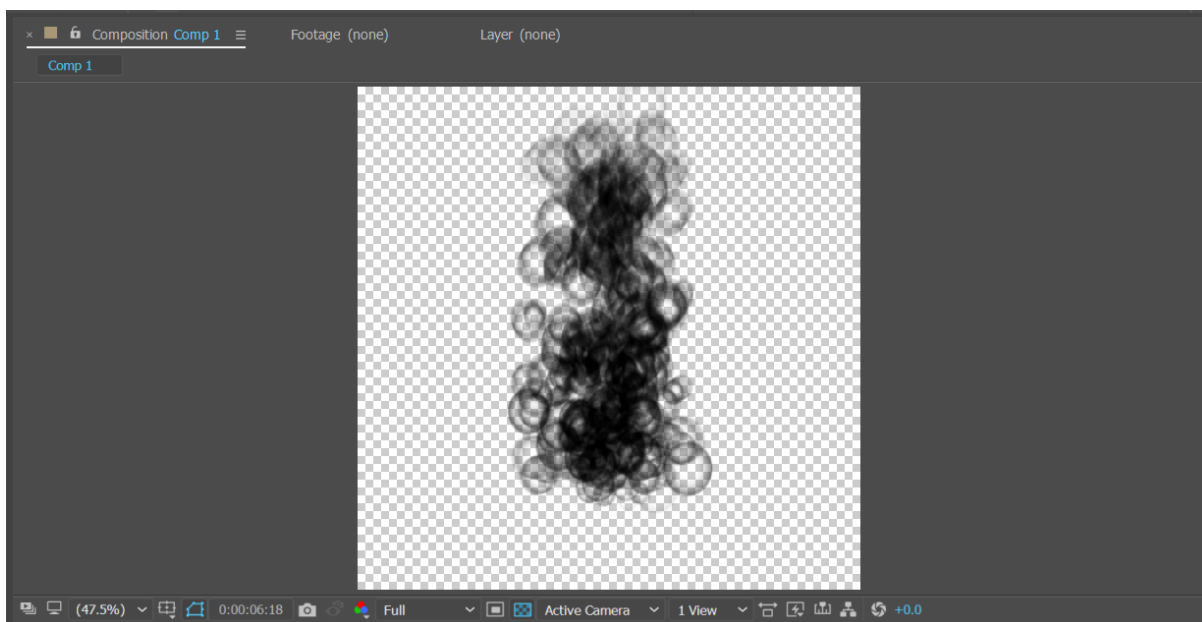


Рисунок 3.23 – Particle Type – Lens Bubble

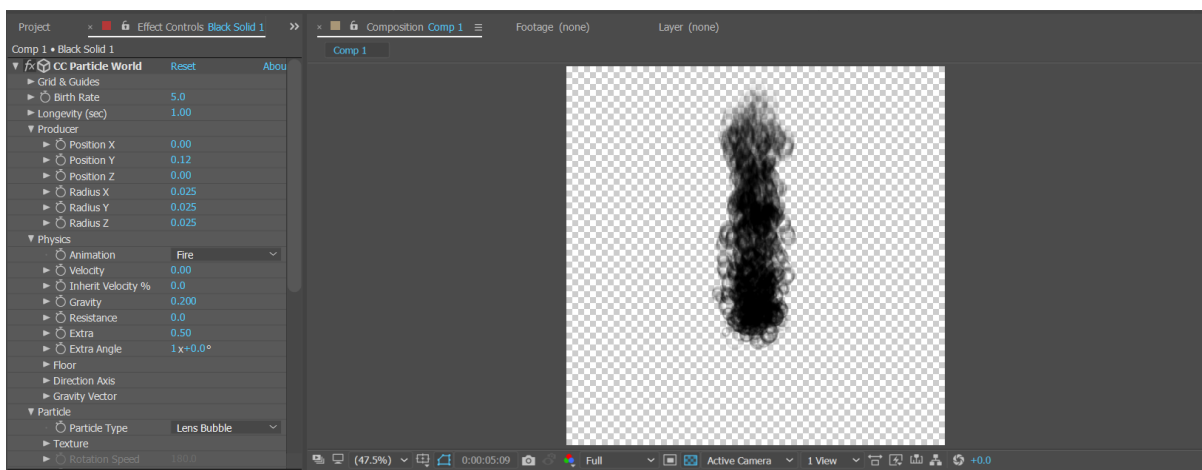


Рисунок 3.24 - Відкоректовані параметри та результат дії ефекту CC Particle World

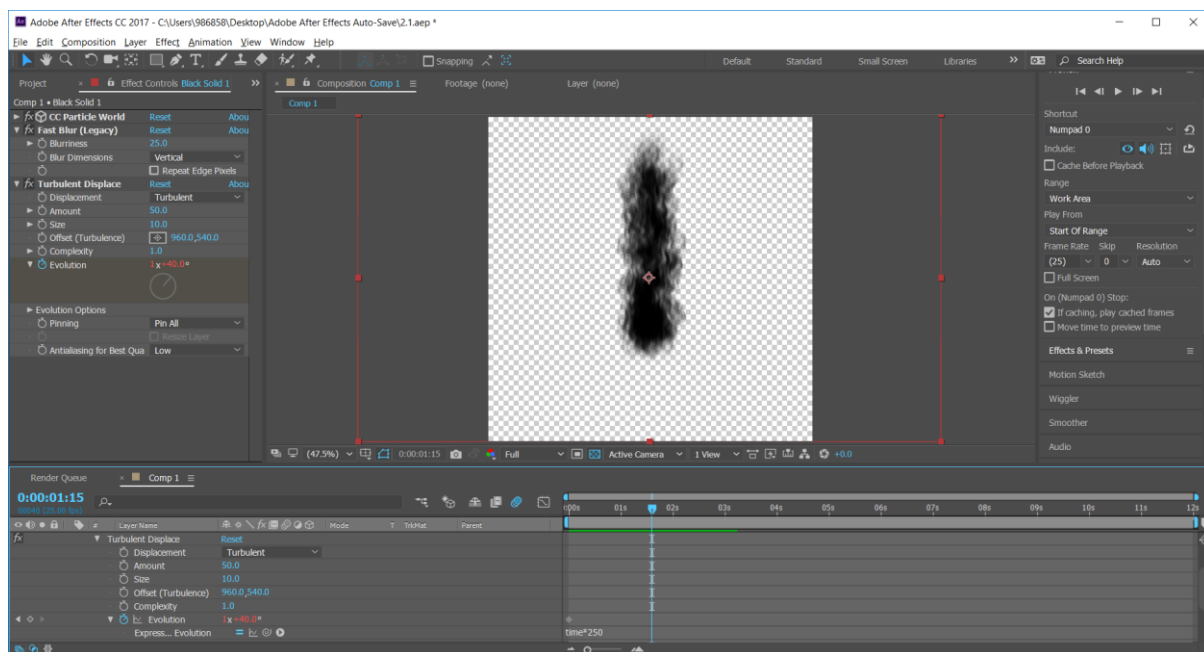


Рисунок 3.25 - Параметри та результат дії ефектів Fast Blur та Turbulent Displace

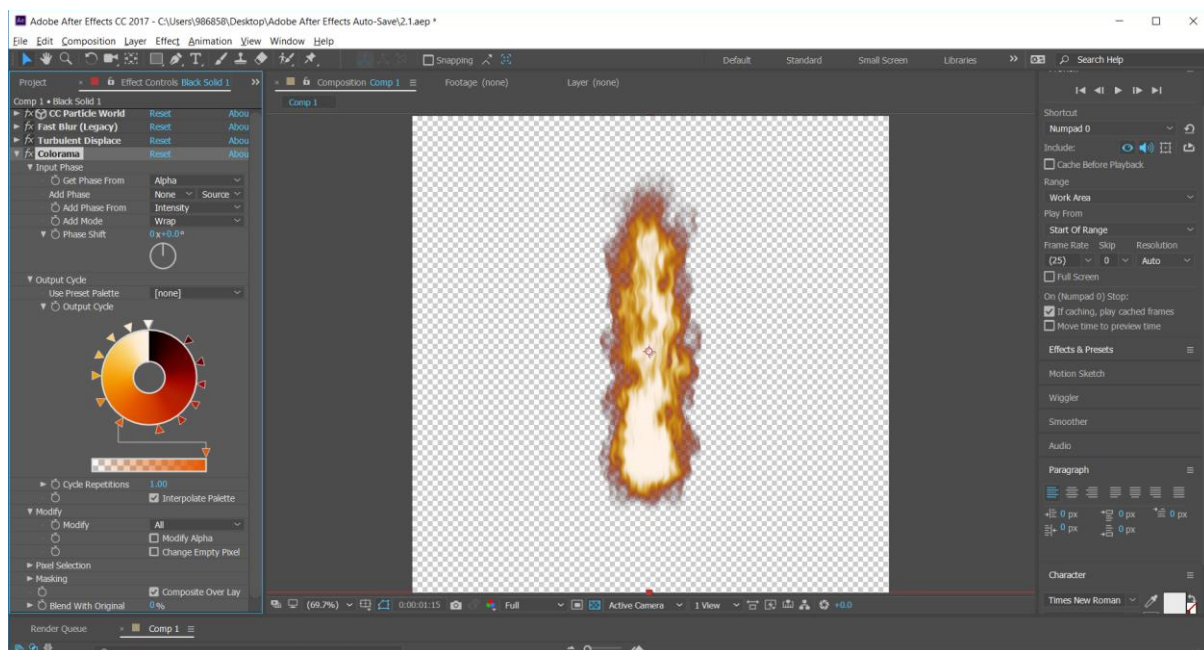


Рисунок 3.26 - Параметри та результат дії ефекту Colorama

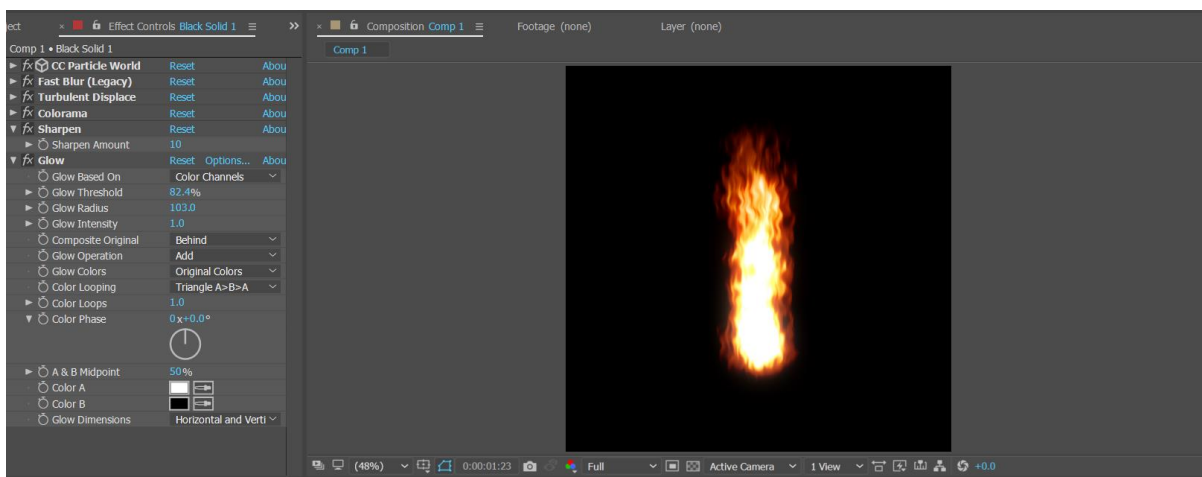


Рисунок 3.27 - Параметри та результат дії ефектів Sharpen та Glow

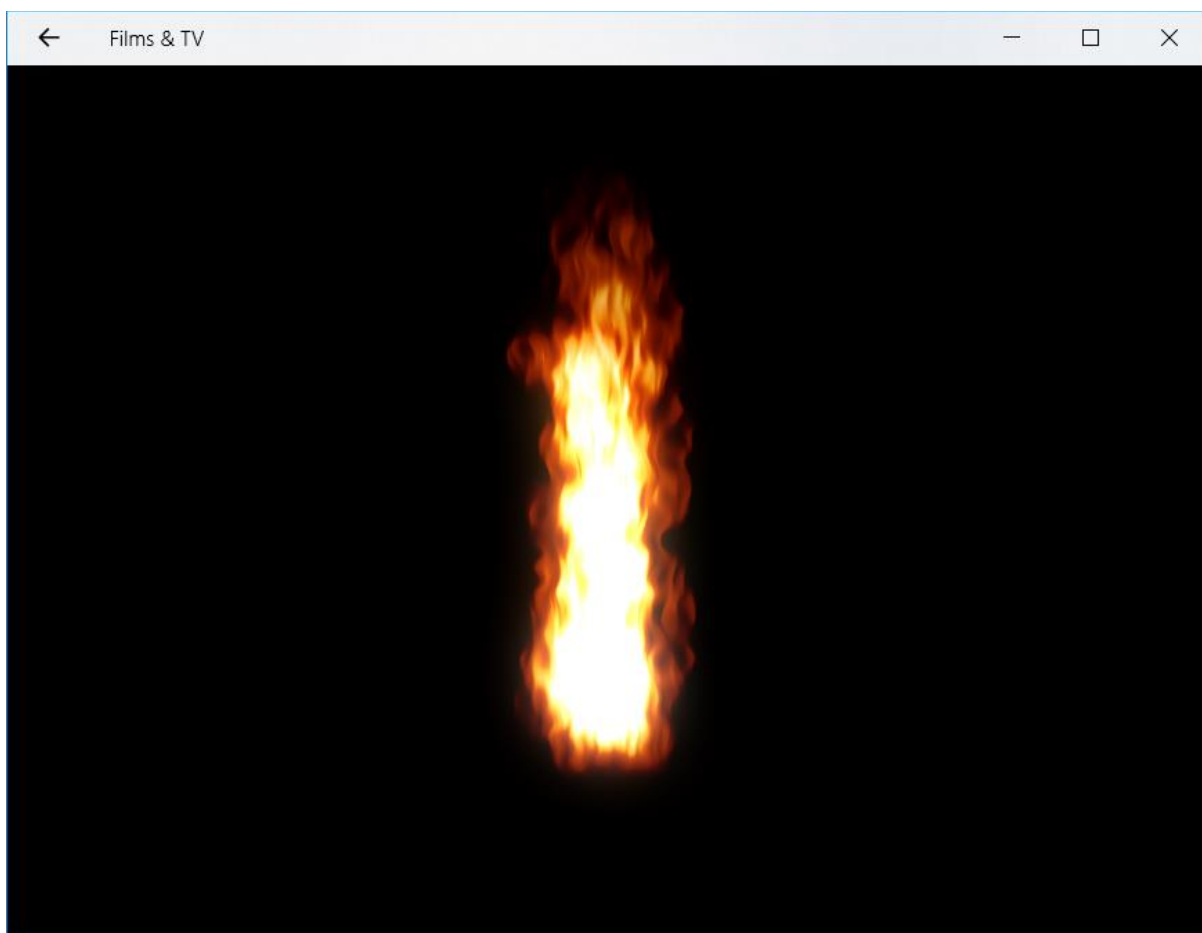


Рисунок 3.28 – Результат рендерінгу

3.3 Створення ефекту полум'я у програмі *After Effects* на основі ефектів Фрактальний шум - Fractal Noise, з допомогою масок та шарів та з використанням системи частинок - CC Particle World

Для створення більш реалістичного ефекту використано маски та додаткові шари. За допомогою інструменту Перо (Pen Tool) створена необхідна форма маски (див. Рис.3.29). Розмиття додається зміною параметра Mask Feather (див. Рис.3.30).

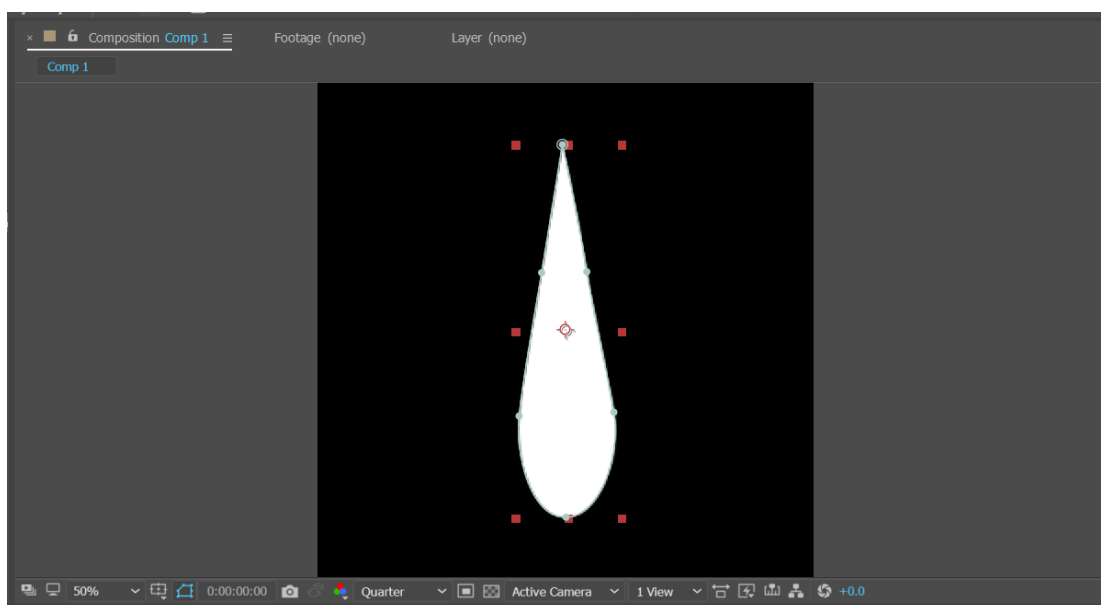


Рисунок 3.29 – Маска

Далі маски дублюються тричі для подальшої роботи - для поступової зміни характеру полум'я.

За допомогою ефекту Криві (Curves) та створивши додаткову маску застосовуємо альфа канал на нижню частину об'єкта (див. Рис.3.31).

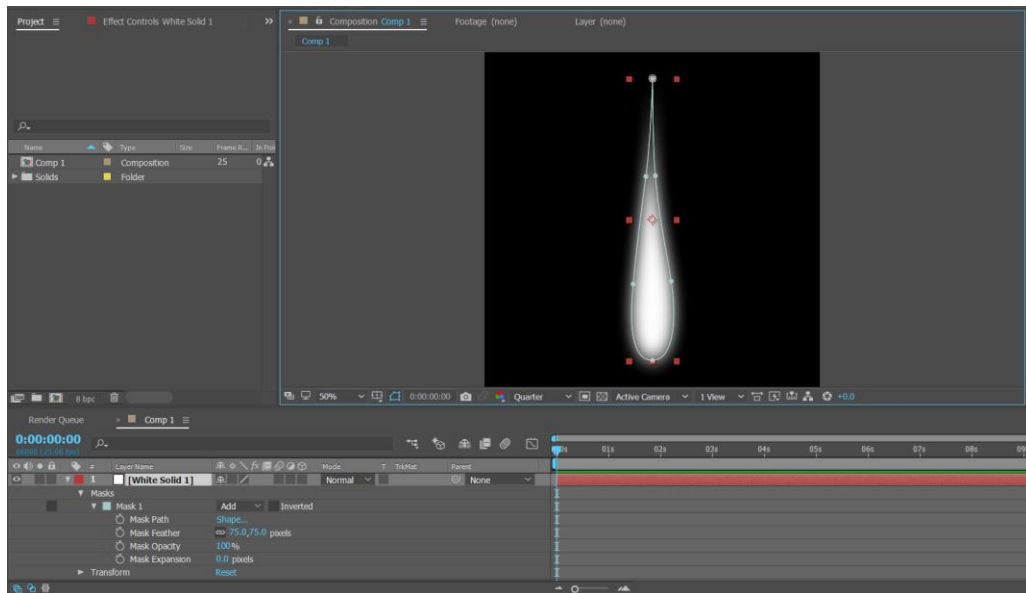


Рисунок 3.30 – Коригування параметрів маски

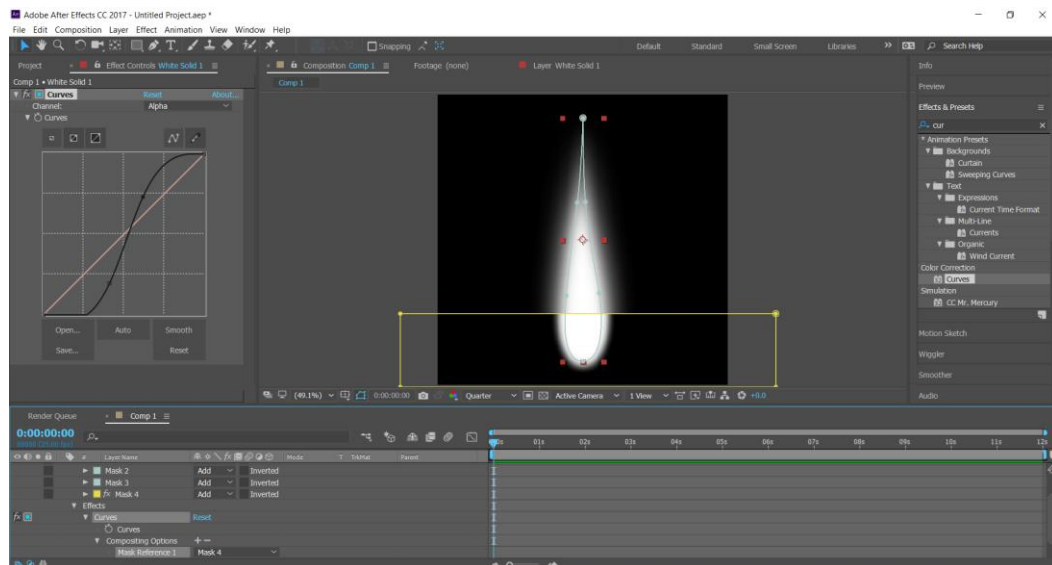


Рисунок 3.31 – Використання альфа каналу

За допомогою ефекту Фрактальний шум (Fractal Noise) та задання функцією параметру Зміщення турбулентності (Offset Turbulence) отримана основа руху полум'я (див. Рис.3.32).

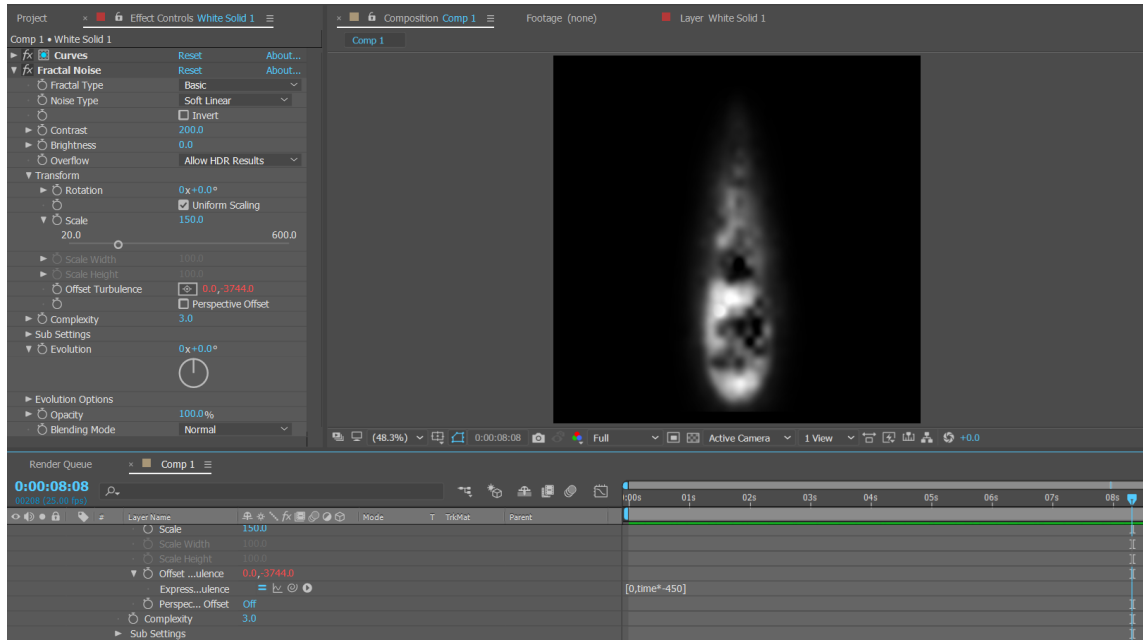


Рисунок 3.32 – Параметри та результат дії ефекту Fractal Noise

Аналогічно до маски нижньої частини об'єкту, створено і параметризовано дві додаткові маски верхньої частини (див. Рис.3.33) але зв'язані з ефектом Фрактальний шум – відповідно, для послідовного шумування. Зайве свічення нейтралізоване ефектом Яскравість та контрастність (Brightness & Contrast) (див. Рис.3.34).

Для створення текстури полум'я також використовується ефект Фрактальний шум. Дублювання цього ефекту з різною швидкістю зміщення забезпечує об'ємне сприйняття анімації (див. Рис.3.35). За допомогою ефекту Деформація сітки (Mesh Warp) змінена форма анімації (див. Рис.3.36), а зайві вигини приховуємо маскою та ефектом Наповнення (Fill) (див. Рис.3.37).

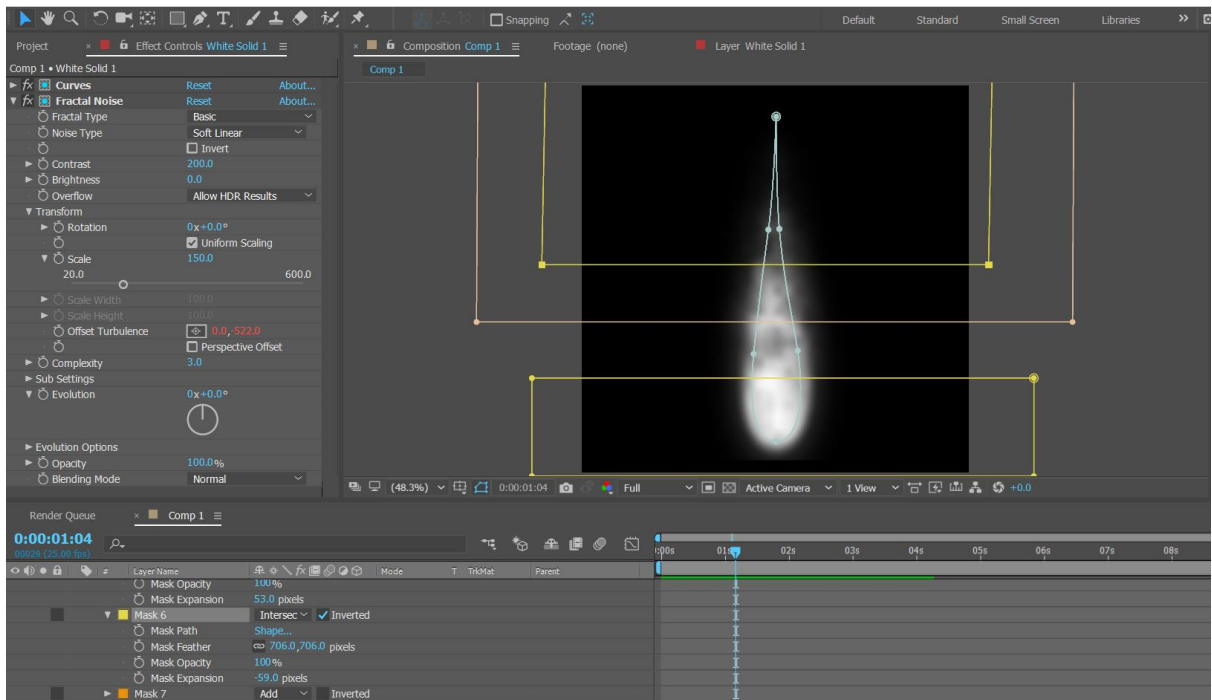


Рисунок 3.33 – Створення масок для верхньої частини об’єкту

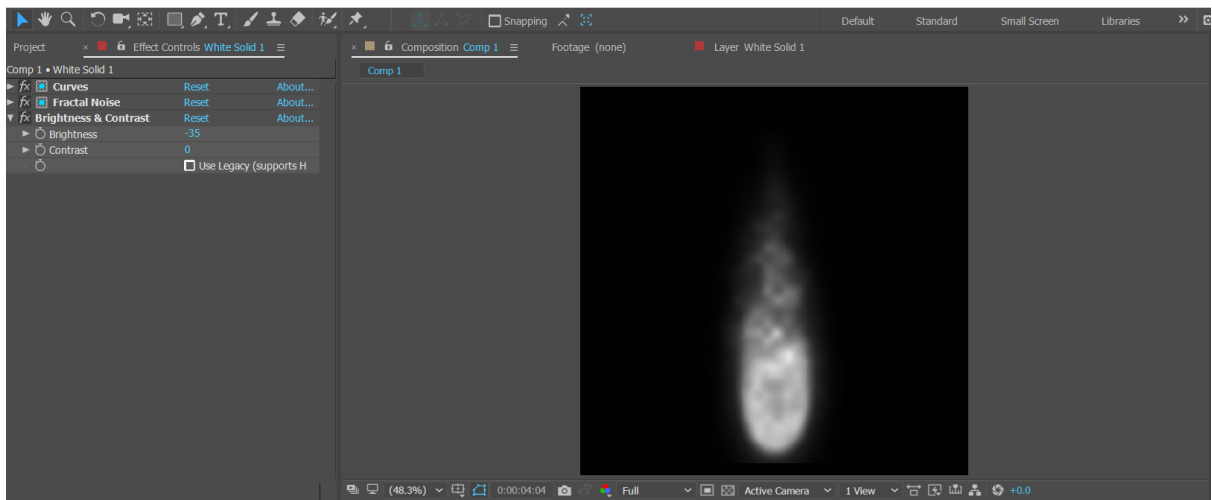


Рисунок 3.34 – Параметри та результат дії ефекту Brightness & Contrast

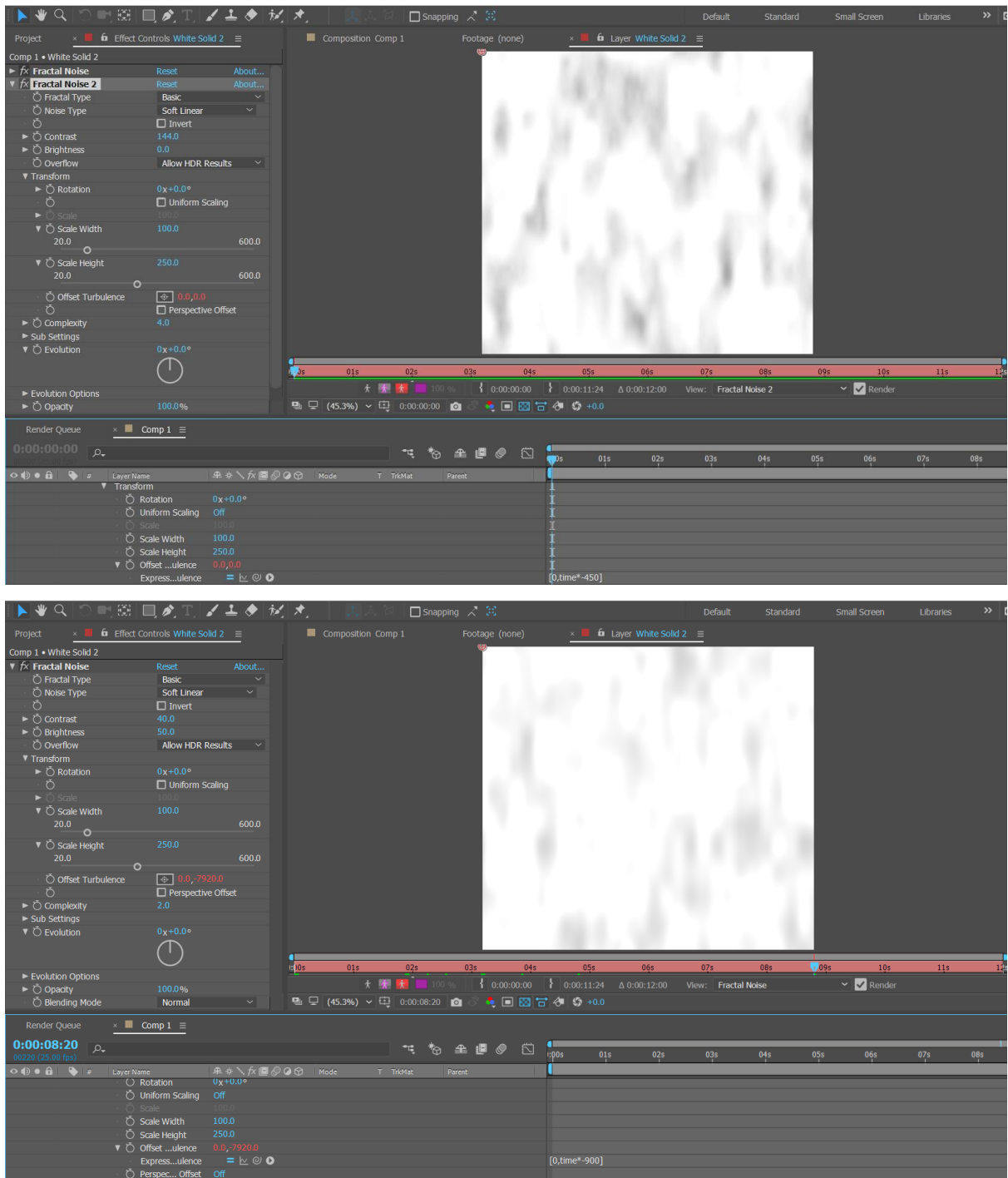


Рисунок 3.35 – Параметри та результат дії ефекту Fractal Noise

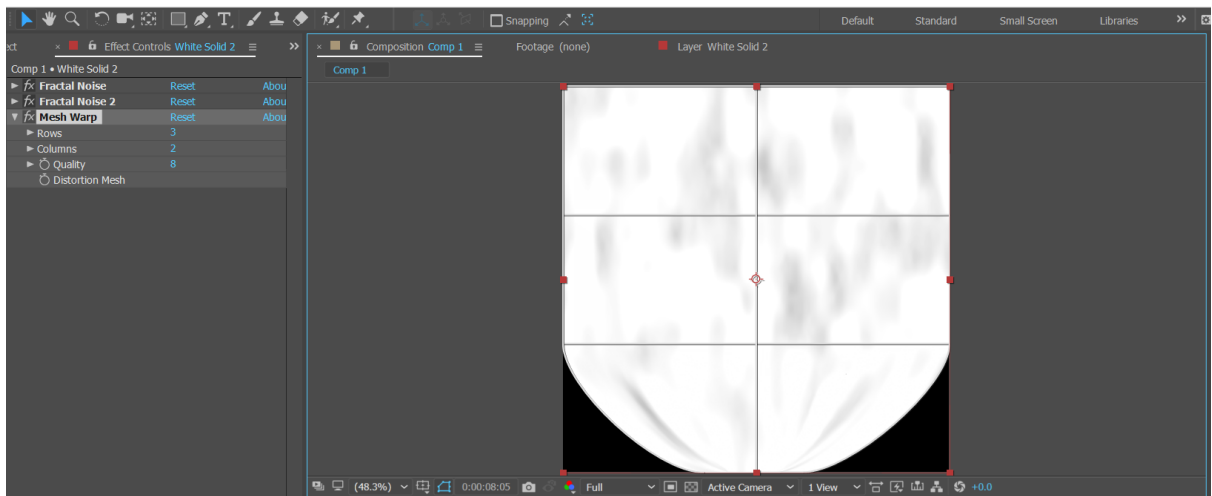


Рисунок 3.36 – Параметри та результат дії ефекту Mesh Warp

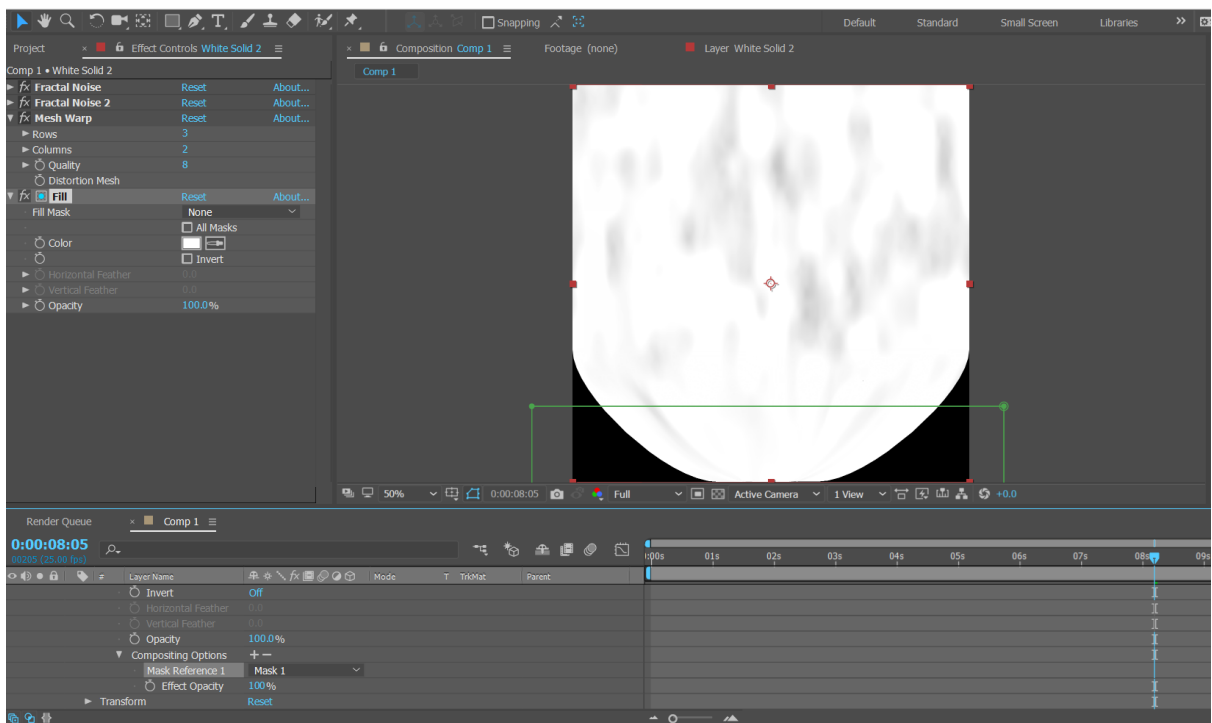


Рисунок 3.37 – Параметри та результат дії ефекту Fill

Додано окремий шар з ефектом Фрактального шуму та вибрано, крім параметра Зміщення, параметр Еволюція (див. Рис.3.38). Також форма коригується та накладається розмиття (див. Рис.3.39).

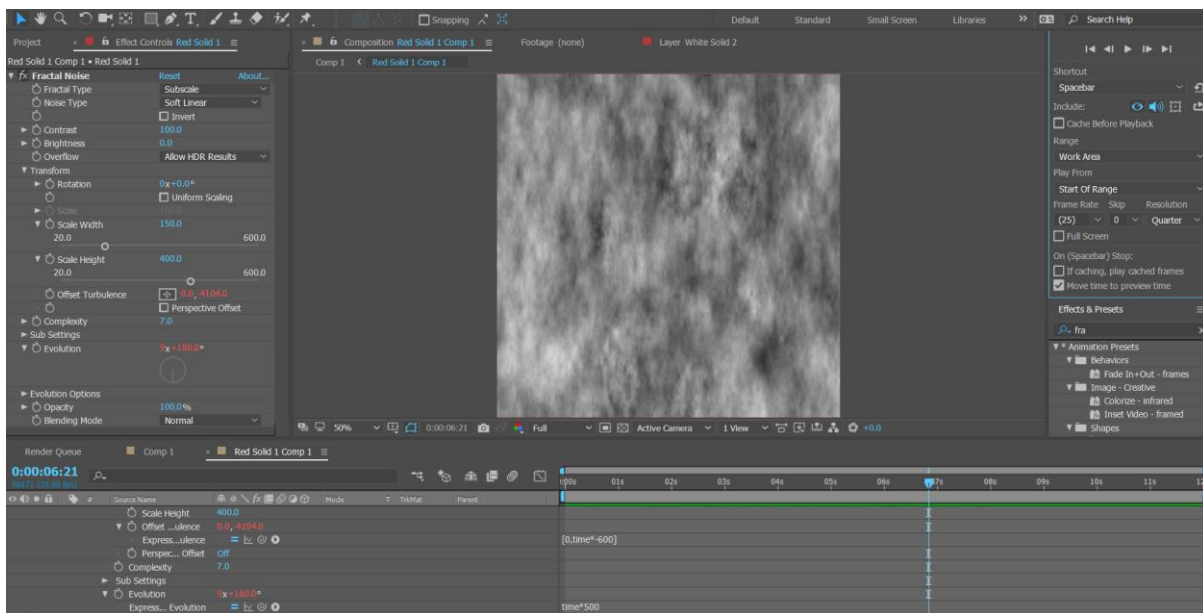


Рисунок 3.38 – Параметри та результат дії ефекту Fractal Noise

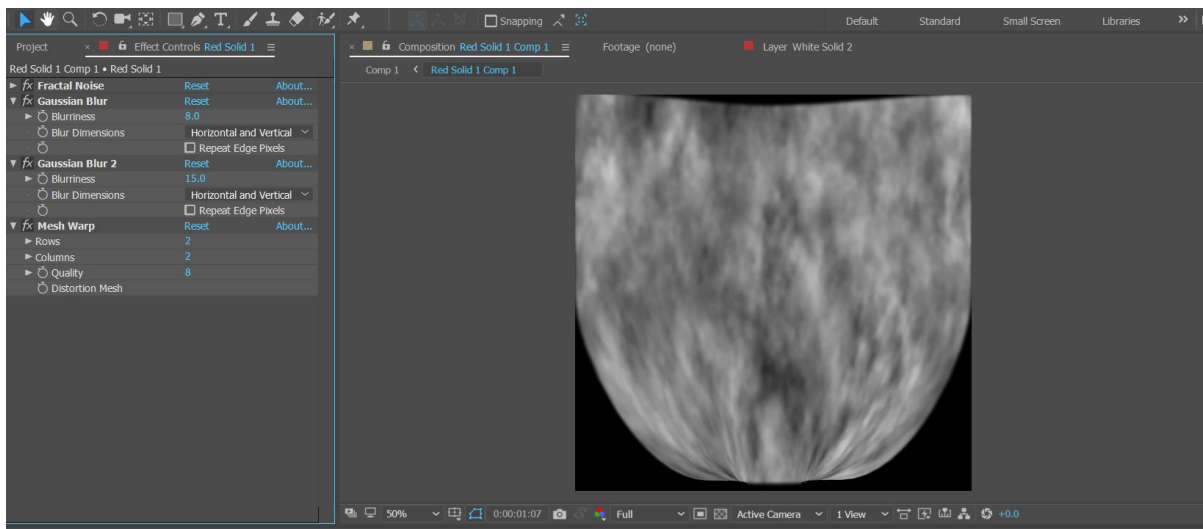


Рисунок 3.39 – Кінцевий вигляд додаткового шару

Інвертуюча маска зменшить деформацію нижньої частини анімації (див. Рис.3.40). Поєднуємо за допомогою ефекту Карта зміщення (Displacement Map) (див. Рис.3.41).

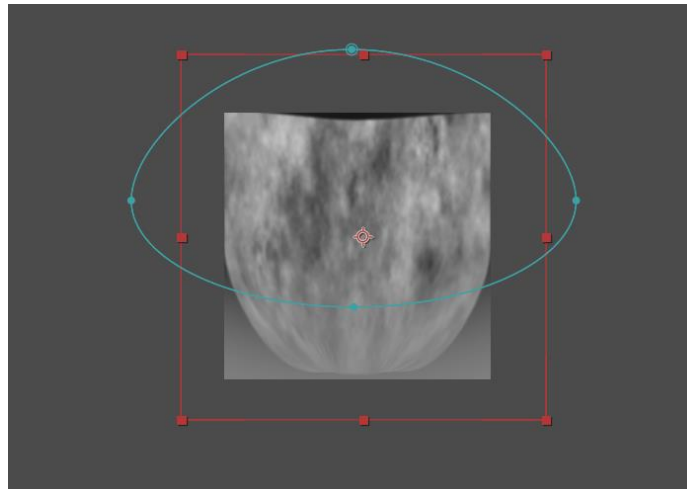


Рисунок 3.40 - Інвертуюча маска

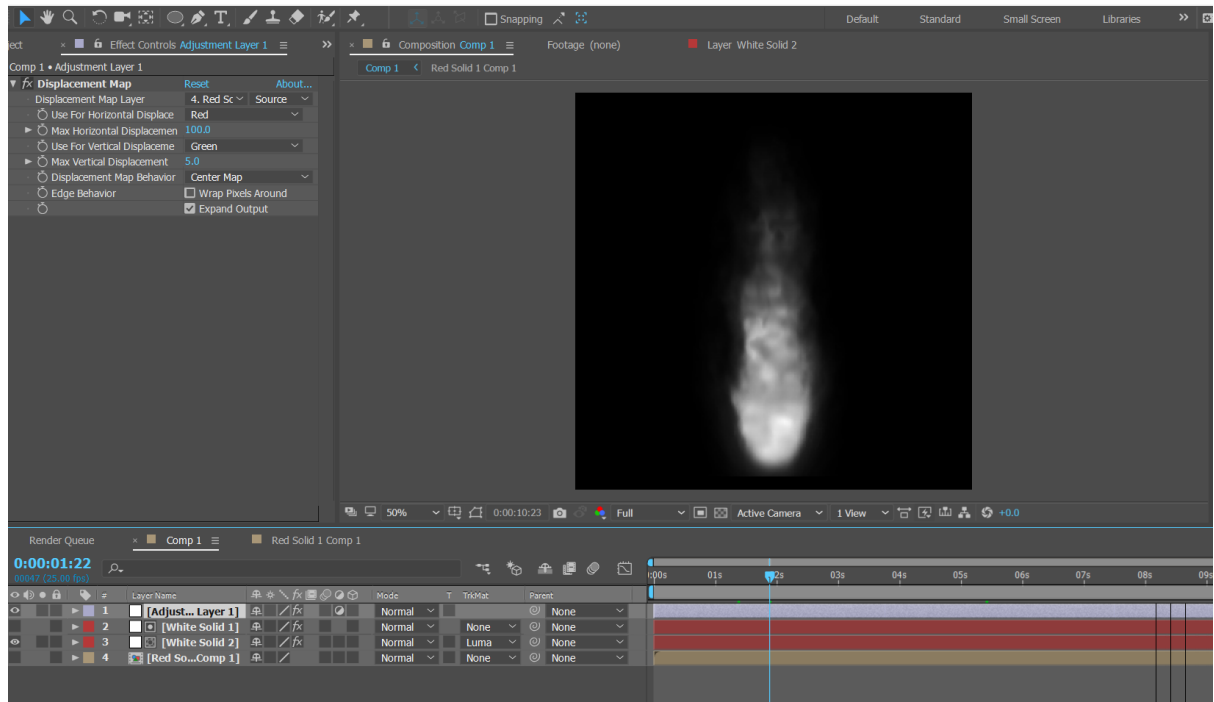


Рисунок 3.41- Результат поєднання

Для фарбування анімації створене нова композиція, яка включає цю і дві її копії, шар фону та безпосередньо шар фарбування (див. Рис.3.42).

Для отримання оптимального кольору задіяні такі ефекти:

- Контурна чіткість (Unsharp Mask)
- Яскравість і контраст (Brightness & Contrast)
- Криві (Curves)

Останній елемент – ефект CC Particle World (див. Рис.3.43).

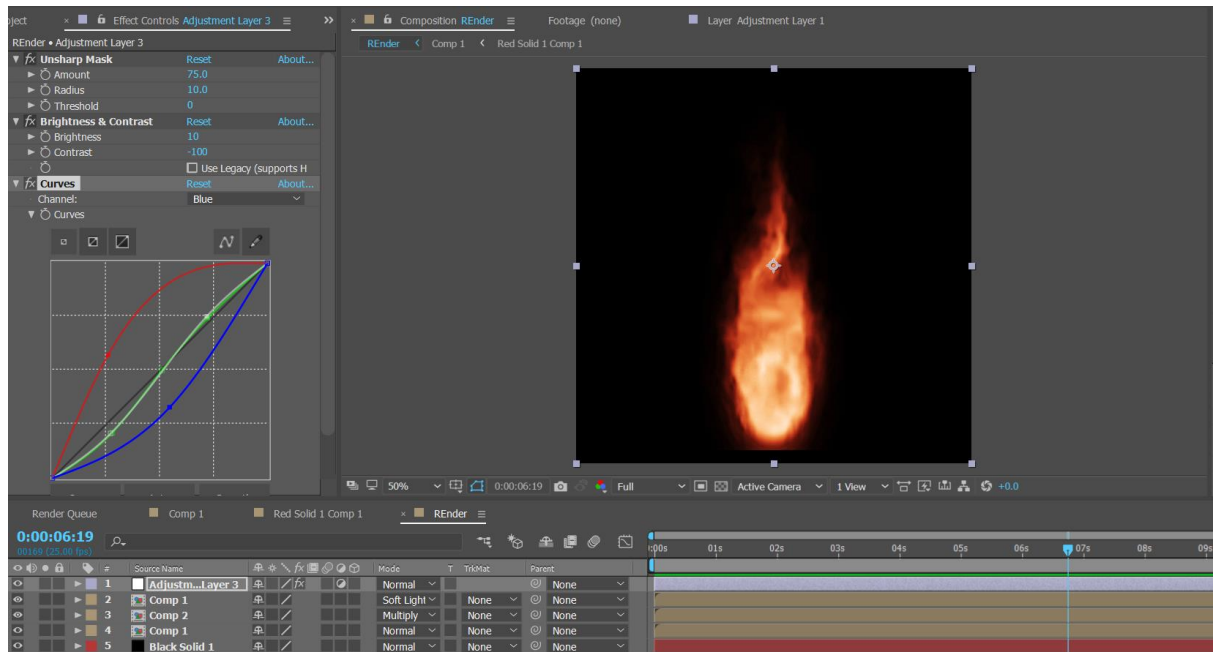


Рисунок 3.42 – Фарбування анімації

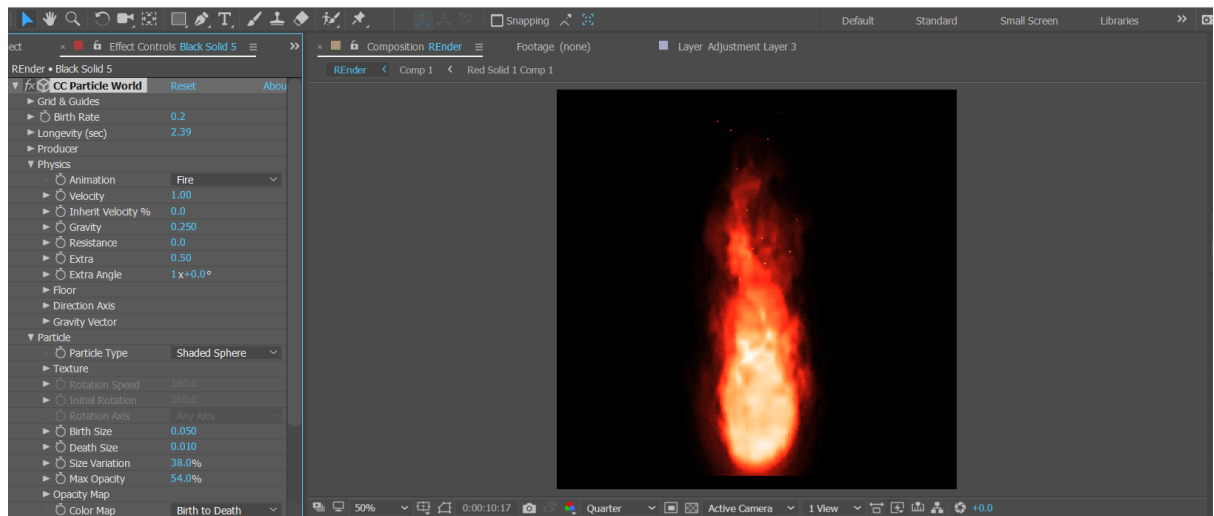


Рисунок 3.43 – Параметри та результат дії ефекту CC Particle World

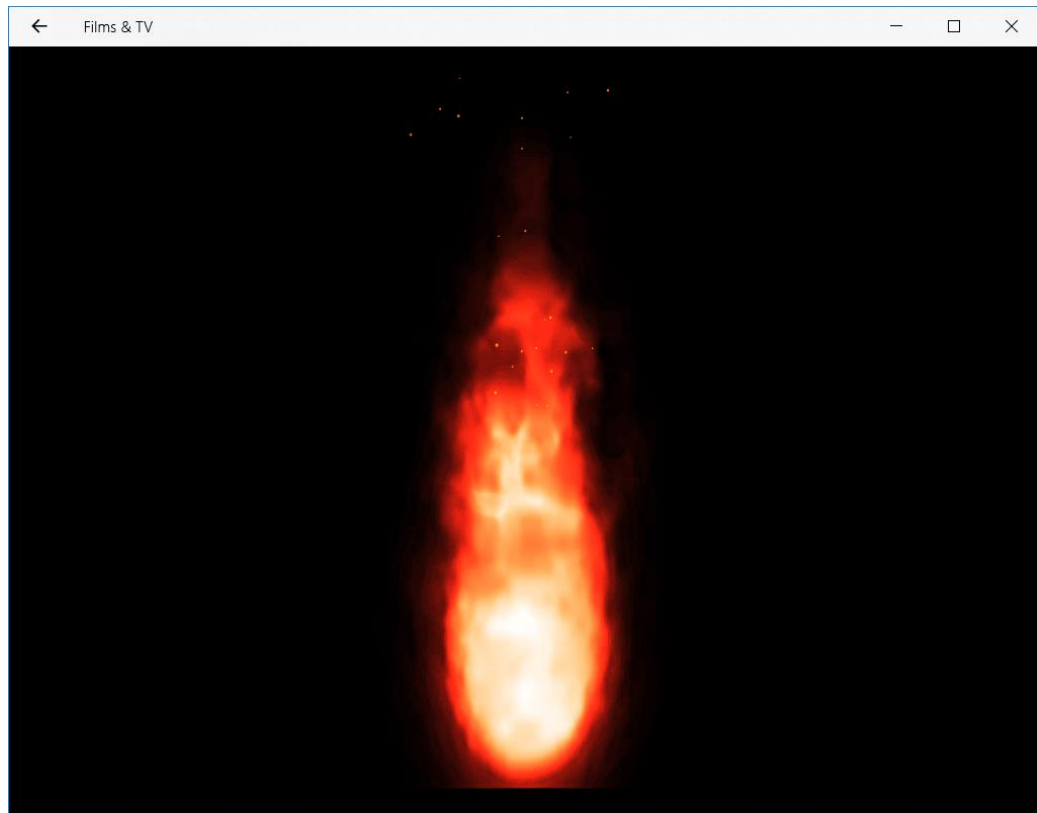


Рисунок 3.44 – Результат рендерінгу

3.4 Створення ефекту полум'я у програмі *Blender*

Для роботи з системою частинок в програмному середовищі *Blender* існує спеціальна панель - Particle System panel (див. Рис.3.45).

Типи:

- Емітер (Emitter) – об'єкт, що джерелом випромінення частинок.
- Волосся (Hair) – випромінення частинок у вигляді ниток, що контролюються.

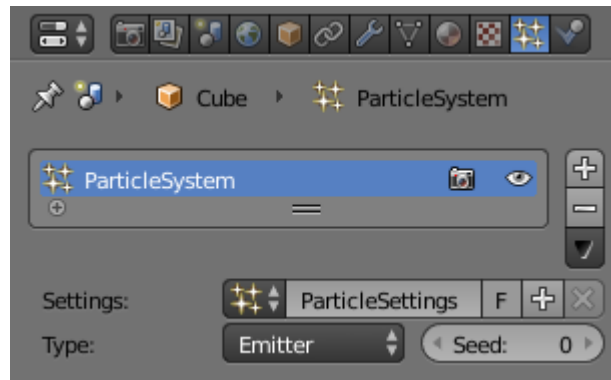


Рисунок 3.45 – Панель управління системою частинок

Стандартне створення ефекту починається з створення об'єкту випромінення, додання системи частинок, однієї або декількох. Налаштування кольору, текстури, фізичних властивостей, анімації, вибору і формування шляху розповсюдження та рендерінг (див. Рис.3.46).

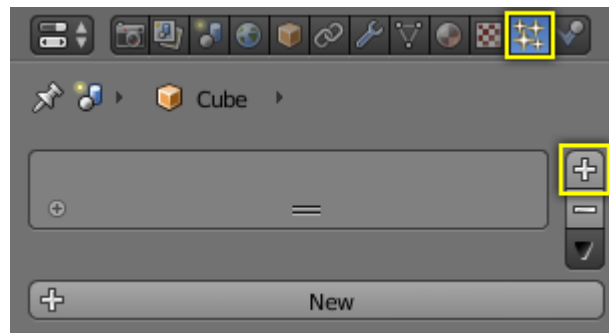


Рисунок 3.46 – Додавання системи частинок

Кожна система частинок має окремі параметри, які застосовують на неї. Ці параметри можуть бути розподілені між різними системами частинок, тому кожен параметр не потрібно копіювати вручну і може використовуватись той самий ефект для декількох об'єктів.

Для роботи в нодовому режимі можливо впливати на систему частинок за допомогою вузлу інформації про частинки (Particle Info). Він може бути використаний

у дереві матеріальних вузлів для об'єктів, які використовуються як об'єкти дублювання (Dupli), коли ви використовуєте режим рендерінгу об'єкта (Object Render) або групи (Group Render) системи частинок (див. Рис.3.47).

Цей вузол дає доступ до даних частинки, яка породила екземпляр об'єкта. Він використовується для внесення певних змін в окремий матеріал, призначений для декількох екземплярів об'єкта дублювання (Dupli).

Виходи:

- Index - Індекс частки (від 0 до кількості частинок).
- Random - Випадкове часткове значення в діапазоні від 0 до 1. Його можна, наприклад, використовувати в комбінації з рампою кольорів, щоб рандомізувати колір частинки.
- Age - Вік частки в кадрах.
- Lifetime - Загальний час життя частинки в кадрах.
- Location - Місцезнаходження частки.
- Size - Розмір частки.
- Velocity - Швидкість частинки.
- Angular velocity - Кутова швидкість частинки.

В програмному середовищі Blender, для створення вогню використаний вбудований ефект Швидкий дим (Quick smoke) (див.3.48).

У вкладці Фізичні властивості (Physics) тип потоку (Type Flow) змінено на полум'я. В нодовому режимі додані блоки для фарбування ефекту (ColorRamp, Emission, Multiply, Attribute) – рисунок 3.49.

Зміна характеру поширення частинок відбувається на вкладці Smoke flames – швидкість, щільність, тривалість, температура, сила та інші (див рис.3.50).



Рисунок 3.47 – Вузол «Інформація про частинки»

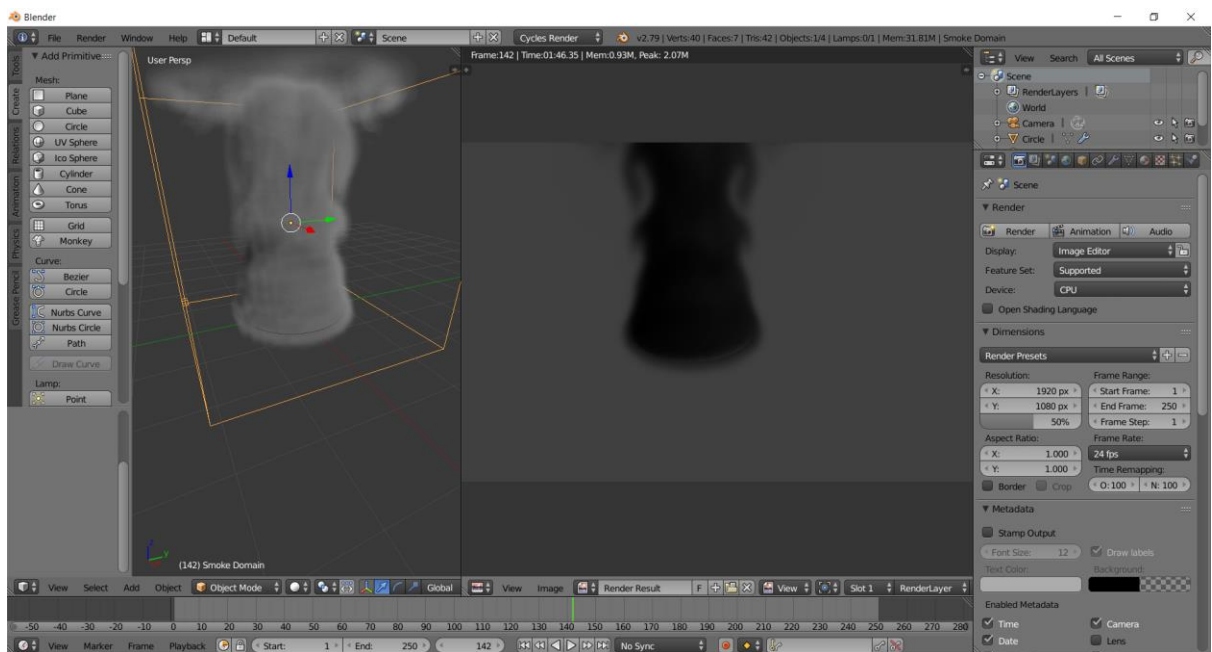


Рисунок 3.48 – Результат ефекту Quick smoke

Для створення окремих язиків полум'я використані текстури. Налаштування текстур знаходиться на вкладці текстури (див. Рис.3.51). На вкладці Фізичні властивості додається зміщення за часом для уникнення однорідності спалахів (див. Рис.3.52).

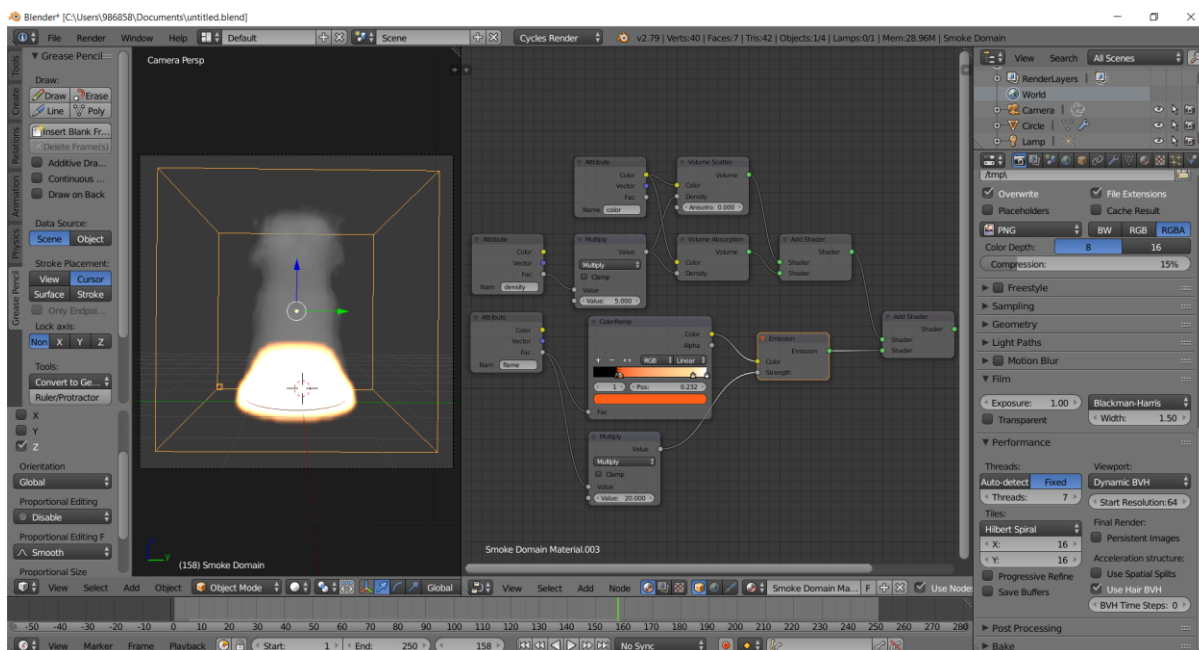


Рисунок 3.49 – Використання нодового режиму для фарбування ефекту

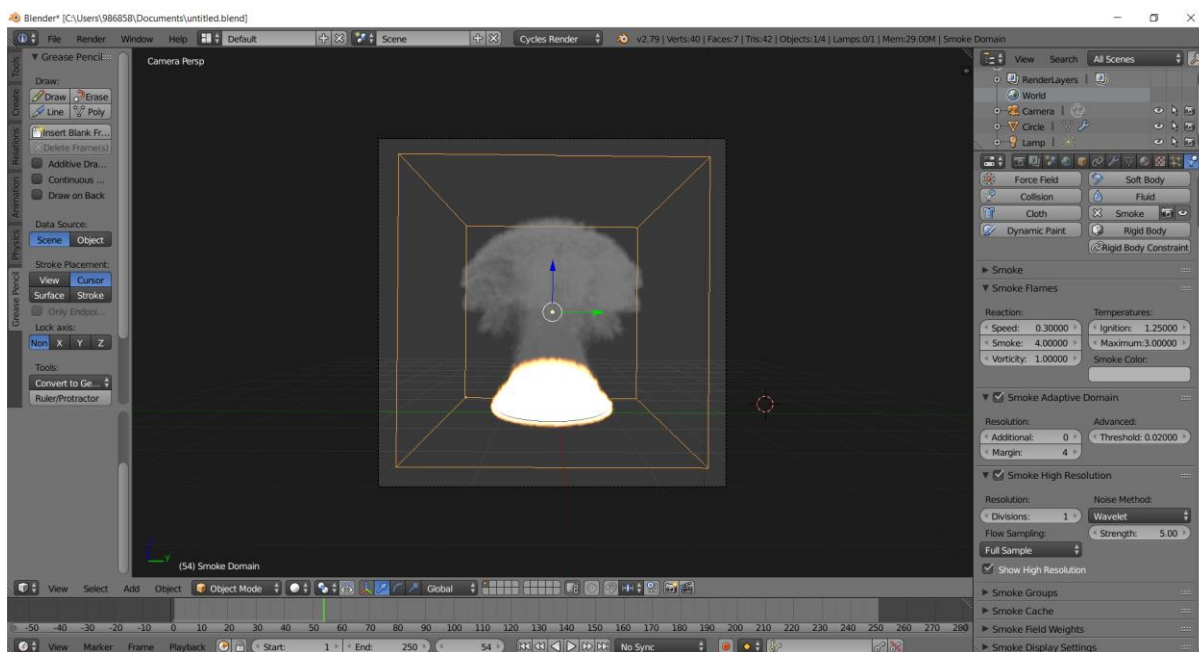


Рисунок 3.50 – Параметри зміни характеру поширення частинок

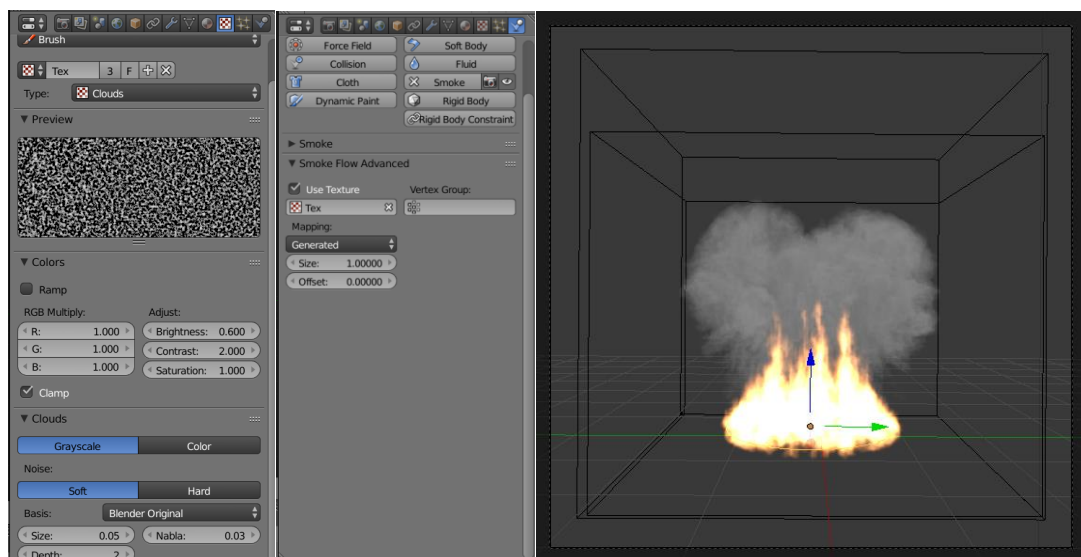


Рисунок 3.51 – Налаштування та результат текстури ефекту.

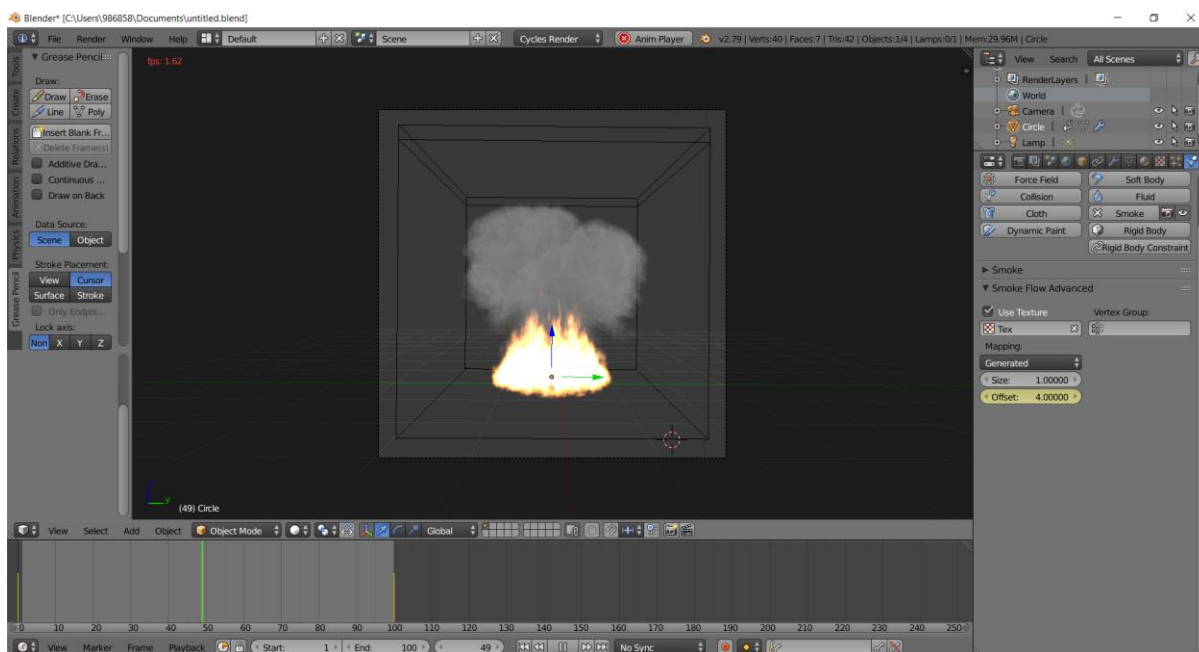


Рисунок 3.52 – Налаштування та результат зміщення.



Рисунок 3.53 – Проміжний результат рендерінгу

Для моделювання іскор застосовується система частинок (див. Рис.3.54). Зміна напрямку розповсюдження досягається застосуванням об'єкту «Поле» (Див. Рис.3.55).

Для зміни кольору частинок у нодовому режимі додані такі блоки, як ColorRamp (для зміни кольору), Dot product (для з'єднання блоку системи частинок з наступними), Multiplay (комбінування блоків зміни кольору), Emission (емісія), Transparent BSDF (додавання сяйва) та Mix Shader (поєднання шейдерів) (див. Рис.3.56).

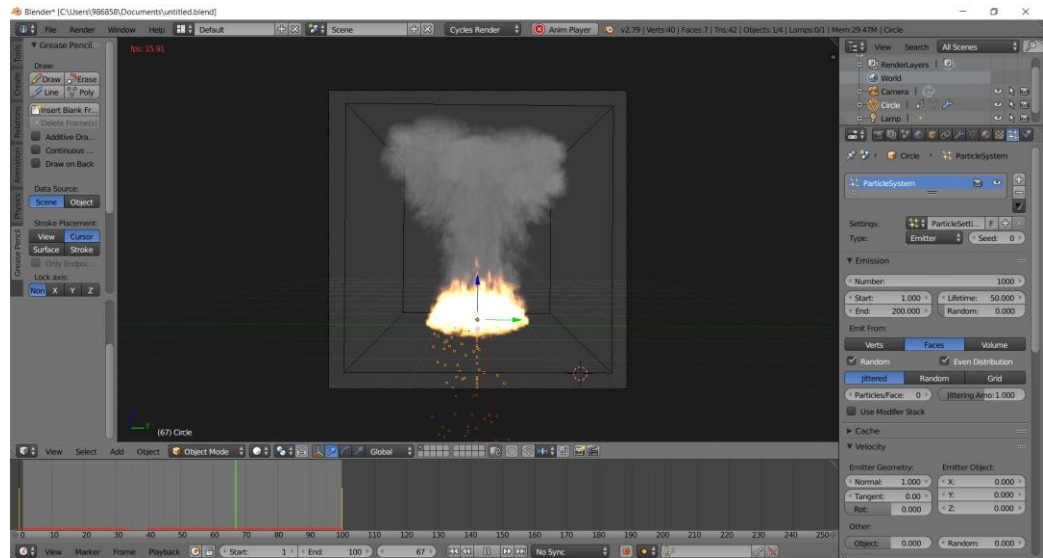


Рисунок 3.54 – Результат та налаштування вкладки «Система частинок»

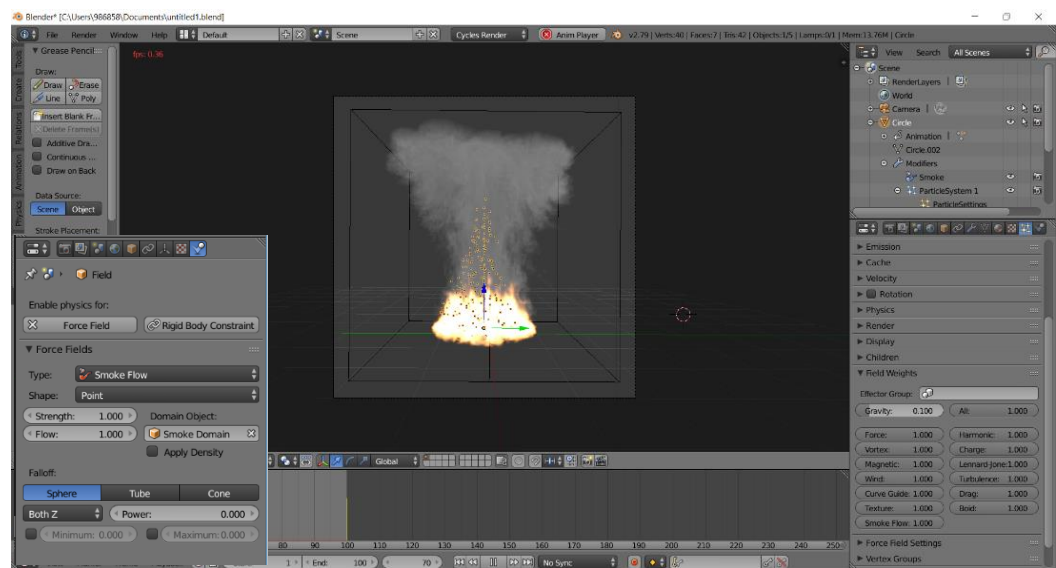


Рисунок 3.55 – Результат та налаштування об'єкту «Поле»

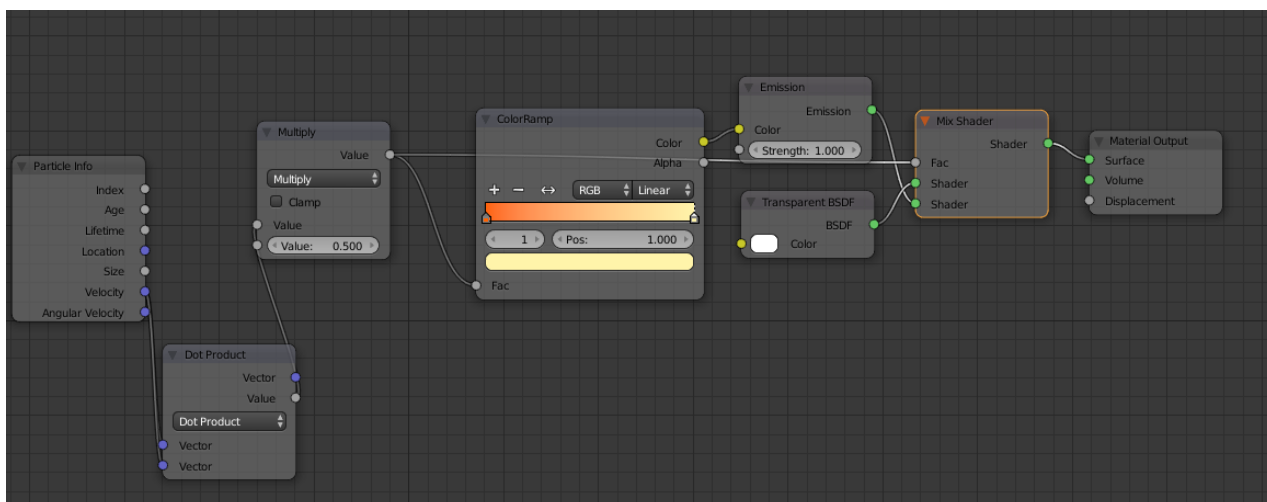


Рисунок 3.56 – Нодова структура системи частинок

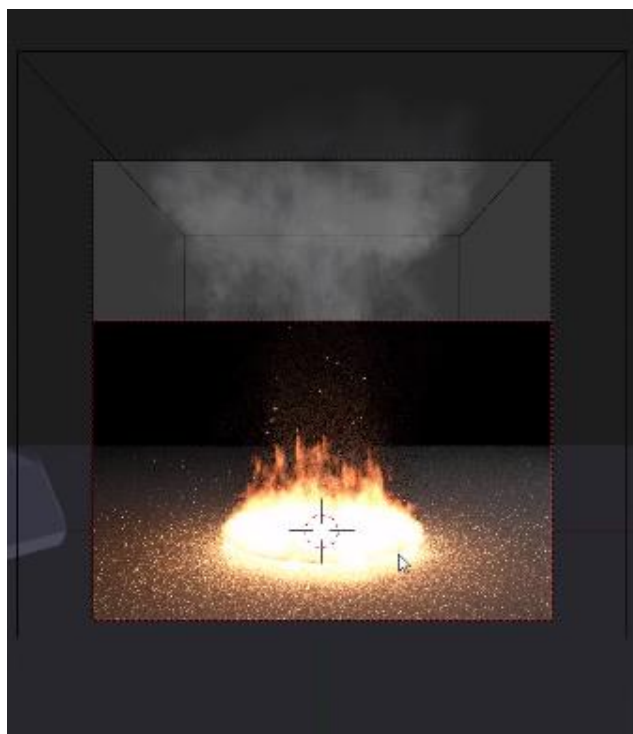


Рисунок 3.57 - Результат рендерінгу

Висновки до розділу

Змодельовані спеціальні візуальний ефекту у вигляді полум'я в програмних середовищах: Magic Particles, Adobe After Effects, Blender. Для створення ефекту при невеликих потужностях з високою реалістичністю доцільно використовувати програму Adobe After Effects, з на основі таких вмонтованих плагінів як Фрактальний шум і Система частинок з комбінацією таких інструментів як шари, маски та текстури.

Сформульовано методику створення полум'я в програмних середовищах: Magic Particles, Adobe After Effects, Blender та створено відеоряд з використанням системи частинок.

Показано, що для створення спеціальних візуальних ефектів полум'я використовують плагіни та програми лише цього ефекту, спеціальні ефекти в універсальних графічних редакторах та комбінації різних ефектів для досягнення необхідних результатів.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Загальні відомості

Загальна суть стартапу полягає в такому – при дослідженні ринку виробництві мультимедійного контенту запропоновано програмне середовище для створення візуальних ефектів. Зміст ідеї та визначення характеристик ідеї стартапу наведено в табл. 4.1 та табл. 4.2.

Таблиця 4.1 – Зміст ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Запропонувати програмне середовище для створення спеціальних ефектів при виробництві мультимедійного контенту	1. Кіновиробництво	Використання нових технологій та спрощення процесу виробництва контенту
	2. Рекламні та презентаційні відеокліпи	Використання сучасних ефектів для зацікавлення клієнтів
	3. Анімації	Поширення високоякісного контенту в соціальних мережах заради підвищення рейтингу сторінки

Таблиця 4.2 – Визначення характеристик ідеї проекту

№ п/п	Техніко- економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропоно ваний метод	Загальнов живаний метод			

1	Пропозиція встановлення та налаштування програмного середовища	Дає змогу	Дає змогу	Розробка та встановлення нового програмного середовища дорогий процес	Підвищення зручності виробництва медіа контенту виправдане очікуванням та витраченими коштами	Рішення є більш ефективним відносно інших
2	Підвищення якості контенту в умовах низької апаратної потужності	Дає змогу	Не дає змогу	Не гарантується 100% гарантія отримання бажаного результату	Потребує додаткових затрат на плагіни	Забезпечується якісне зображення

4.2 Технологічний аудит ідеї проекту

У таблиці 4.3 показано оцінку технологічної здійсненності ідеї проекту та наведено технології, що можуть бути використані для реалізації проекту.

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Організація виробництва мультимедійного контенту	Налаштування персонального комп'ютера	Наявна	При обмеженому бюджеті недоступна

2		Застосування програмних систем	Наявна	Доступна
3		Використання додаткових плагінів	Наявна	При обмеженому бюджеті недоступна

Обрана технологія реалізації ідеї проекту: застосування програмних систем.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

У таблиці 4.4 показано попередню характеристику потенційного ринку стартап-проекту.

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку(найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од	300000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Зацікавлення потенційних клієнтів
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	$300000/210000 = 143\%$

У таблиці 4.5 показано характеристику потенційних клієнтів стартап-проекту.

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
-------	--------------------------	--	---	-----------------------------

1	Забезпечення простого програмного середовища для виробництва мультимедійного контенту	Медійні компанії, користувачі соціальних мереж, маркетингові компанії	Рівень очікування якості мультимедійного контенту	Відповідність результату найвищим стандартам якості
2	Забезпечення якісного контенту при невеликих потужностях	Медійні компанії, користувачі соціальних мереж, маркетингові компанії	Кожна з потенційних цільових груп має свої вимоги до стандартів відеозображення	Забезпечення якості мультимедійного контенту від рівня потреб споживача

У табл. 4.6 показані фактори загроз реалізації стартап-проекту.

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Внесення додаткових сервісних послуг
2	Втрата конкуренції	Втрата рангу надійного поставника	Якісне та кількісне нарощування інтенсивності та грамотна цінова політика

У табл.4.7 показано фактори можливостей при реалізації стартап-проекту.

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
-------	--------	------------------	--------------------------

1	Перехід до домінування на ринку медійних послуг	Зростання попиту	Якісне та кількісне нарощування потужностей
2	Імплементация технологій в існуючі програмні середовища	Зростання попиту внаслідок зростання кількості користувачів	Якісне та кількісне нарощування потужностей

У таблиці 4.8 визначено особливості конкурентного середовища та його вплив на впровадження проекту[15].

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Конкуренція	Використання вже існуючих технологій	Стандартизація на високому рівні
2. Локальний	Відсутність єдиного постачальника послуг	Окремий підхід до кожної локальної ділянки
3. Міжгалузева	Відсутня	Відсутня
4. Товарно-видова	Застосування стандартизованих технологій	За необхідності, використання загальноновживаних апаратних та програмних засобів
5. Цінова	Застосування спеціалізованих програмних середовищ, які мають значну ціну	Можливості заощадити за допомогою застосування загальноновживаних апаратних засобів
6. Марочна	Кожна діагностика має бути стандартизованою	Отримання переваги на ринку медійних послуг

У таблиці 4.9 показано аналіз конкуренції проекту в галузі за М. Портером.

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Постачальники програмного забезпечення	Необхідність пошуку постачальників	Залучення малопопулярних постачальників	Незалежність у прийнятті клієнтських рішень	Надання переваги більш авторитетним функціональним рішенням
Висновки :	Середня	Можливість виходу на ринок є	Постачальники диктують цінову політику на додатки та додаткові функції	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови від діагностики

У табл. 4.10 показано фактори конкурентоспроможності та їх обґрунтування.

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурси на покращення якості дослідження виробництва мультимедійного контенту
2	Надання сервісних послуг	Сервісна підтримка апаратної та програмної частини

У табл. 4.11 наведено сильні та слабкі сторони проекту.

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
			-3	-2	-1	0	+1	+2	+3
1	Раціональніший ціновий показник	15		+					
2	Надання сервісних послуг	12					+		
3	Періодична діагностика	7			+				
4	Необхідність залучення висококваліфікованих кадрів	7						+	

У табл.4.12 наведено SWOT-аналіз стартап-проекту.

Таблиця 4.12 - SWOT- аналіз стартап-проекту

Сильні сторони: раціональний ціновий показник, надання сервісних послуг	Слабкі сторони: періодична діагностика, необхідність залучення висококваліфікованих кадрів
Можливості: Перехід до використання стабільного програмного середовища. Імплементація методу в існуючі медіа комплекси	Загрози: Незацікавленість клієнтів, Втрата монополії

Альтернативи ринкового впровадження стартап-проекту наведені у табл.4.13.

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Укладення договорів з маркетинговими та медійними компаніями і соц мережами та швидке захоплення ринку при використанні нового рішення	висока	незначні
2	Використання сервісів загального вжитку для підвищення конкурентоспроможності	середня	незначні

Обрана альтернатива - укладення договорів з маркетинговими, медійними компаніями та соціальними мережами, а також швидке захоплення ринку при використанні нового рішення.

4.4 Розроблення ринкової стратегії проекту

Обґрунтування вибору цільових груп потенційних споживачів наведено у табл. 4.14 [16].

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Маркетингові та медійні компанії Надання якісних послуг	Середня	Високий	Середня	Середня
2	Індивідуальні відео-блогери	Низька	Середній	Середня	Висока

Визначення базової стратегії розвитку наведено у табл. 4.15.

Таблиця 4.15 - Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Використання альтернативних технологій	Встановлення нового стандарту якості	Зацікавлення та залучення гігантів у медійній галузі	Стратегія диференціації

2	Дешевизна проекту	Раціональніші витрати на створення та подальша підтримка	Застосування загальноновживаних програмних рішень замість спеціалізованих комплексів	Стратегія лідерства по витратах
---	-------------------	--	--	---------------------------------

Визначення базової стратегії конкурентної поведінки наведено у табл.4.16.

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Забирати існуючих та шукати нових	Не буде	Стратегія виклику лідера

Визначення стратегії позиціонування наведено у табл. 4.17.

Таблиця 4.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока якість послуг	Стратегія диференціації	Новизна, гарант якості, точність дослідження	Якість, надійність, точність
2	Мінімальні витрати	Стратегія лідерства по витратах	Універсальність запропонованого рішення	Дешевизна, універсальність

4.5 Розроблення маркетингової програми стартап-проекту

Ключові переваги концепції потенційного товару наведено у табл. 4.18.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода яку пропонує товар	Ключові переваги перед конкурентами
1	Якість	Висока якість, надійність	Надійність
2	Дешевизна	Раціональне використання коштів, дешевше обладнання	Дешевизна

Визначено три рівні моделі товару. Сутність та складові рівнів товару наведено у табл. 4.19.

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Якісний товар та послуги, стандартизована якість послуг		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1)Вартість обслуговування, 2)Кількість комплектів обладнання 3)Строк безвідмовної праці 4)Технологічна собівартість товару	1) М 2) М 3) М 4) М	1)Е 2) Пр 3)Нд 4)Тх

	Якість: міжнародні стандарти якості, постійна підтримка обладнання
	Доставка, встановлення та налаштування
	Марка: Відео-виробництво
III. Товар із підкріпленням	До продажу – обладнання, встановлення
	Після продажу – сервісна підтримка

За рахунок чого потенційний товар буде захищено від копіювання: специфічна методика обробки даних і контроль якості відео-трансляції.

Визначення меж встановлення ціни на послугу наведено у табл. 4.20.

Таблиця 4.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	10000 у.о./од. (стандартна методика)	-	Середній	Н.5000 у.о. – В.11000 у.о. (Товар) Н.300 у.о. – В.1000 у.о. (Послуга)

Формування системи збуту послуги наведено у табл. 4.21.

Таблиця 4.21 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Орієнтована на отримання максимальної якості власної трансляції	Поставки якісного, точного та надійного товару	Значна	Договірна система збуту

Концепції маркетингових комунікацій наведено у табл. 4.22.

Таблиця 4.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Зацікавленість в якісному та точному продукті з раціональним використанням ресурсів	Мережні ресурси	Гарантованість якості та стандартизація, політика сервісності	Зацікавити у покращеннях пов'язаних із зростаючою популярністю послуг	Представлення центру синхронізації відправною точкою на шляху до над якісного контенту
2	Зацікавленість у великій кількості продукту із дотриманням умов якості	Мережні ресурси	Глибина каналу постачальників, гарант якості	Зацікавити у позитивних сторонах первісності та в глибині каналу постачання	Представлення послуг центру синхронізації єдиним раціональним шляхом у забезпеченні якісного інструменту медіа виробництва

Висновки до розділу

1. Комерціалізацію стартап-проекту щодо розвитку та впровадження запропонованого програмного середовища для виробництва мультимедійного контенту, можна вважати доцільною. На дану пропозицію на ринку медійних послуг присутній попит, наразі він задовольняється програмами замінниками та більш дорогими рішеннями, саме тому важливо зайняти нішу конкурента у якості поставника вигідного продукту, порівнюючи з конкурентами. Рентабельність на ринку послуг насамперед обумовлена заміною повної апаратної залежності на універсальність, що обумовлена використанням не спеціалізованих комплексів, а загальноживаного програмного та апаратного забезпечення.

2. Впровадження є перспективним, адже основними групами клієнтів є масштабні телевізійні компанії, інтернет-компанії та індивідуальні відео-блогери, маркетингові компанії, соціальні мережі і після набуття достатньої авторитетності можливе охоплення у масштабах міжнародних ринків. Конкурентоспроможність проекту обумовлена меншою ціною на повний продукт та високою якістю проведення трансляції в умовах, коли конкуренти за цим параметром у даних умовах програють. Це вигідно вирізняє запропоноване рішення, власне, і є основним критерієм входження на ринок.

3. Обраною альтернативою впровадження було обрано – пошук альтернативних програмних рішень для виробництва мультимедійного контенту. Імплементація проекту доцільна, оскільки рентабельність та зацікавленість потенційних груп клієнтів створює досить сприятливі умови для розвитку проекту.

ВИСНОВКИ

В магістерській дисертації проаналізовано особливості застосування спеціальних ефектів при виробництві мультимедійного контенту. Для вирішення виявлених проблем запропоновано використання програмного середовища Adobe After Effects та метод системи частинок, які дозволяють вирішити проблему моделювання нерегулярних об'єктів та проблему оптимізації та використання малих потужностей.

Досліджено теоретичні аспекти розвитку та використання спеціальних ефектів.

Проаналізовано технології спеціальних ефектів в кіновиробництві а також наявні програмні рішення для створення комп'ютерної графіки такі як 3Ds Max, After Effect, Maya, Cinema 4D, Modo, Houdini, Softimage, LightWave, Blender.

Досліджено алгоритми створення моделі класу нечітких об'єктів в частині генерації, динаміки, візуалізації, ієрархії частинок.

Змодельовані спеціальні візуальний ефекту у вигляді полум'я в програмних середовищах: Magic Particles, Adobe After Effects, Blender. Для створення ефекту при невеликих потужностях з високою реалістичністю доцільно використовувати програму Adobe After Effects, з на основі таких вмонтованих плагінів як Фрактальний шум і Система частинок з комбінацією таких інструментів як шари, маски та текстури.

Сформульовано методику створення полум'я в програмних середовищах: Magic Particles, Adobe After Effects, Blender та створено відеоряд з використанням системи частинок.

Надано порівняльну характеристику застосування цих програмних продуктів.

Розроблено план-стартап проекту для поширення оптимального програмного середовища. При впровадженні стартапу доцільно обрати альтернативу дослідження

поведінки споживачів, здійснити пошук наукових ресурсів, розробити необхідне технічне обладнання, налагодити зв'язки з рекламою, постійно взаємодіяти з покупцями задля перевірки ефективності програмного забезпечення, а також безперервного зворотного зв'язку. Подальше виконання проекту вважається доцільним.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Harada, Takahiro et al. Real-Time Particle-Based simulation on GPUs, In Proc. of SIGGRAPH '07 Special Interest Group on Computer Graphics and Interactive Techniques Conference, Article No.: 52, January 2007, San Diego, CA, United States
2. Leslie Iwerks, Industrial Light & Magic: Creating the Impossible. (television series), 2010, United States
3. James F. Blinn, Light reflection functions for simulation of clouds and dusty surfaces, Proceedings of the 9th annual conference on Computer graphics and interactive techniques, p.21-29, July 26-30, 1982, Boston, Massachusetts, United States
4. C. Csuri , R. Hackathorn , R. Parent , W. Carlson , M. Howard, Towards an interactive high visual complexity animation system, Proceedings of the 6th annual conference on Computer graphics and interactive techniques, p.289-299, August 08-10, 1979, Chicago, Illinois, United States
5. Alain Fournier , Don Fussell , Loren Carpenter, Computer rendering of stochastic models, Communications of the ACM, v.25 n.6, p.371-384, June 1982
6. Luna, Frank D. Introduction to 3D Game Programming with DirectX 9.0c: A Shader Approach./Luna, Frank D// Plano, TX: Wordware Publishing - 2006
7. William T Revees, "Particle systems – A technique for modeling a class of fuzzy objects", ACM Computer Graphics, 1983, pp. 359-376
8. Jim X.CHEN, Xiaodong Fu, and Edward J. Wegman, "Real-Time simulation of dust behaviour generated by a fast traveling vehicle", ACM Transactions on Modeling and Computer Simulation, April 1999, pp. 81-104
9. Alan Norton, Generation and display of geometric fractals in 3-D, Proceedings of the 9th annual conference on Computer graphics and interactive techniques, p.61-67, July 26-30, 1982, Boston, Massachusetts, United States

- 10.Ebert D S, “Advanced Modeling Techniques for Computer Graphics”, ACM Computing Surveys, 1996, 28(1): 153~156.
- 11.PBS. Carl Sagan's Cosmos Series. (television series), Public Broadcasting System, 1980.
- 12.Liao H S, Ho T C, Chuang J H, Lin C C, “Fast rendering of dynamic clouds”. Computers & Graphics, 2005, 29(1): 29~40.
- 13.Shi Jiandi, Jiang Yuming,”The simulation of dynamic cloud based on the fractal method”, Computer Simulation, 2006, 23(4):197~200.
- 14.Huang Tianyun, Zhang Chuanwu, “The fractal interpolation algorithm in fractal simulation of natural scenery”, Computer engineering and design, 2007, 28(16): 3994~3997.
- 15.Тиль П. От нуля к единице: как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
- 16.Харниш В. Правила прибыльных стартапов: как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва: Манн, Иванов и Фербер, 2012. – 279 с.

ДОДАТОК А

Реферат англійською мовою на тему магістерської дисертації

ABSTRACT

Modelling phenomena such as clouds, smoke, water, and fire has proved difficult with the existing techniques of computer image synthesis. These "fuzzy" objects do not have smooth, well-defined, and shiny surfaces; instead their surfaces are irregular, complex, and ill defined. We are interested in their dynamic and fluid changes in shape and appearance. They are not rigid objects, nor can their motions be described by the simple affine transformations that are common in computer graphics.

This paper presents a method for the modelling of fuzzy objects that we call particle systems. The representation of particle systems differs in three basic ways from representations normally used in image synthesis. First, an object is represented not by a set of primitive surface elements, such as polygons or patches, that define its boundary, but as clouds of primitive particles that define its volume. Second, a particle system is not a static entity. Its particles change form and move with the passage of time. New particles are "born" and old particles "die." Third, an object represented by a particle system is not deterministic, since its shape and form are not completely specified. Instead, stochastic processes are used to create and change an object's shape and appearance.

In modelling fuzzy objects, the particle system approach has several important advantages over classical surface-oriented techniques. First, a particle (for now, think of a particle as a point in three-dimensional size) is a much simpler primitive than a polygon, the simplest of the surface representations. Therefore, in the same amount of computation time one can process more of the basic primitives and produce a more complex image. Because a particle is simple, it is also easy to motion-blur. Motion-blurring of fast-moving objects for the removal of temporal aliasing effects has been largely ignored in computer image synthesis to date. A second advantage is that the model definition is procedural and is controlled by random numbers. Therefore, obtaining a highly detailed model does not necessarily require a great deal of human design time as is often the case with existing

surface-based systems. Because it is procedural, a particle system can adjust its level of detail to suit a specific set of viewing parameters. As with fractal surfaces [5], zooming in on a particle system can reveal more and more detail. Third, particle systems model objects that are "alive," that is, they change form over a period. It is difficult to represent complex dynamics of this form with surface-based modelling techniques.

Modelling objects as collections of particles is not a new idea. Fifteen years ago, the earliest computer video games depicted exploding spaceships with many little glowing dots that filled the screen. Point sources have been used as a graphics data type in many three-dimensional modelling systems (e.g., the early Evans and Sutherland flight simulators), although there are few real references to them in the literature. Roger Wilson at Ohio State used particles to model smoke emerging from a smokestack. There were neither stochastic controls nor dynamics in his model. Alvy Ray Smith and Jim Blinn used particles to model star creation and death in galaxies for the Cosmos series. Alan Norton used particles to generate and display three-dimensional fractal shapes. Jim Blinn discussed light reflection functions for simulating light passing through and being reflected by layers of particles. His technique was used to produce images of the rings of Saturn. Blinn did not address the fuzzy object modelling problem which is the topic of this paper. Volumetric representations have also been proposed as viable alternatives to surface representations. Solid modelling is a form of volumetric representation, as is the work of Norm Badler and Joe O'Rourke on "bubble- man". The use of stochastic modelling relates our work to the recent advances in fractal modelling.

A particle system is a collection of many minute particles that together represent a fuzzy object. Over a period of time, particles are generated into a system, move and change from within the system, and die from the system.

To compute each frame in a motion sequence, the following sequence of steps is performed:

- (1) new particles are generated into the system,
- (2) each new particle is assigned its individual attributes,
- (3) any particles that have existed within the system past their prescribed lifetime are extinguished,
- (4) the remaining particles are moved and transformed according to their dynamic attributes, and finally
- (5) an image of the living particles is rendered in a frame buffer.

The particle system can be programmed to execute any set of instructions at each step. Because it is procedural, this approach can incorporate any computational model that describes the appearance or dynamics of the object. For example, the motions and transformations of particles could be tied to the solution of a system of partial differential equations, or particle attributes could be assigned on the basis of statistical mechanics. We can, therefore, take advantage of models which have been developed in other scientific or engineering disciplines.

Particles are generated into a particle system by means of controlled stochastic processes. One process determines the number of particles entering the system during each interval of time, that is, at a given frame. The number of particles generated is important because it strongly influences the density of the fuzzy object.

For each new particle generated, the particle system must determine values for the following attributes:

- initial position,
- initial velocity (both speed and direction),
- initial size,
- initial color,
- initial transparency,
- shape,

- lifetime.

Several parameters of a particle system control the initial position of its particles. A particle system has a position in three-dimensional space that defines its origin. Two angles of rotation about a coordinate system through this origin give it an orientation. A particle system also has a generation shape which defines a region about its origin into which newly born particles are randomly placed. Among the generation shapes we have implemented are: a sphere of radius r , a circle of radius r in the x - y plane of its coordinate system, and a rectangle of length l and width w in the x - y plane of its coordinate system. Figure 1 shows a typical particle system with a spherical generation shape. More complicated generation shapes based on the laws of nature or on chaotic attractors have been envisioned but not yet implemented.

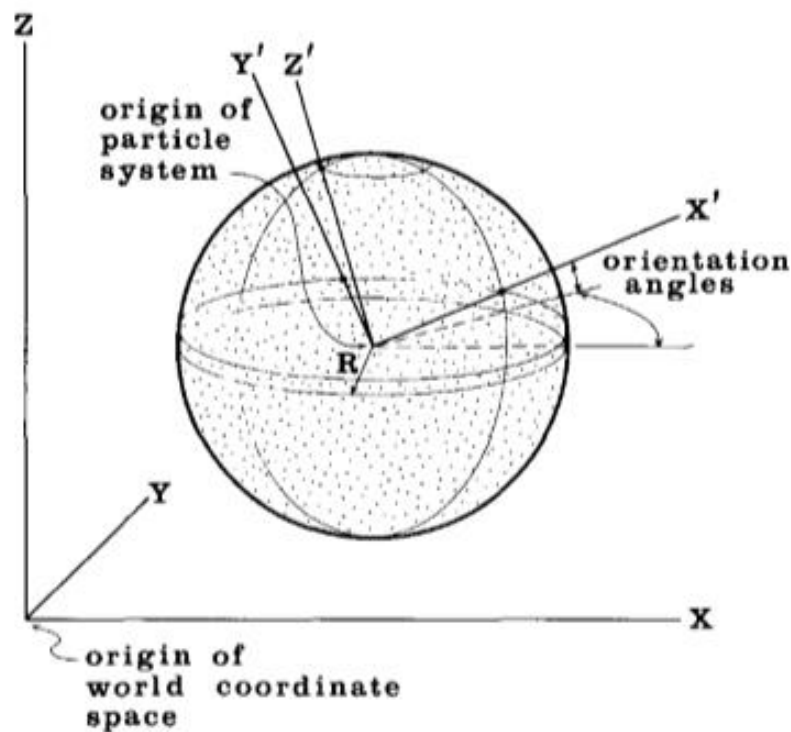


Figure 1 - Typical particle system with spherical generation shape

The generation shape of a particle system also describes the initial direction in which new particles move. In a spherical generation shape, particles move outward away from the origin of the particle system. In a circular or rectangular shape, particles move upward from the x-y plane, but are allowed to vary from the vertical according to an "ejection" angle, which is another parameter (see Figure 2).

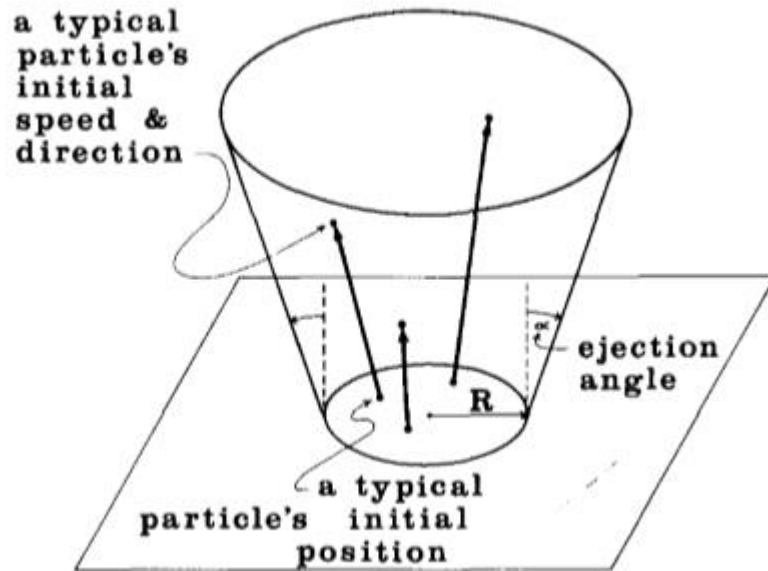


Figure 2 - Form of an explosion-like particle system

A particle system has a parameter that specifies the shape of each of the particles it generates. The particle shapes implemented so far are spherical, rectangular, and streaked spherical.

Particles, especially when modeled as point light sources or as streaks of light, have proved efficient to render--they are merely antialiased lines. Because they are so simple, they lend themselves to a hardware or firmware implementation. With a hardware antialiased line-drawing routine, the computation of our wall- of-fire element would have been two to three times faster.

Particle systems are procedural stochastic representations controlled by several global parameters. Stochastic representations are capable of producing minute detail without requiring substantial user design time. The textures in the fire sequence could not have been modeled with other existing methods. Fire images, scanned in from a photograph or painted, could have been texture mapped, but they would still have been static. Another advantage of a procedural representation is its ability to adapt to several different viewing environments. For example, procedural representations can generate only as much detail as is needed in a frame, potentially saving significant amounts of computation time.